Media Forensics
Fake/Forged/Tempered Multimedia Content Detection

By

Inam Ur Rehman

Malik MatiUllah

Muhammad Musab

Talal Arif Shah

Supervised by:

Prof. Dr. Haider Abbas

Submitted to the faculty of Department of Electrical Engineering,
Military College of Signals, National University of Sciences and Technology, Islamabad,
in partial fulfillment for the requirements of B.E Degree in Electrical Engineering.

June 2023

بسم الله الرحمن الرحيم

In the name of ALLAH, the Most benevolent, the Most Courteous

CERTIFICATE OF CORRECTNESS AND APPROVAL

This is to officially state that the thesis work contained in this report

"Media Forensics – Fake/Forged/tempered Multimedia Content Detection"

is carried out by

Inam Ur Rehman

Malik MatiUllah

Muhammad Musab

Talal Arif Shah

under my supervision and that in my judgement, it is fully ample, in scope and excellence, for the degree of Bachelor of Electrical (Telecom.) Engineering in Military College of Signals, National University of Sciences and Technology (NUST), Islamabad.

Approved by

Supervisor

Prof. Dr. Haider Abbas

Department of EE, MCS

Date: ___________

DECLARATION OF ORIGINALITY

We hereby declare that no portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

ACKNOWLEDGEMENTS

Allah Subhan’Wa’Tala is the sole guidance in all domains.

Our parents, colleagues and most of all supervisor, Prof. Dr. Haider Abbas without your guidance.

The group members, who through all adversities worked steadfastly.

Plagiarism Certificate (Turnitin Report)

This thesis has 12% similarity index. Turnitin report endorsed by Supervisor is attached.

Signature: ___________________

Inam Ur Rehman

NUST Serial no: 300631

Signature: ___________________

Malik MatiUllah

NUST Serial no: 301843

Signature: ___________________

Muhammad Musab

NUST Serial no: 288785

Signature: ___________________

Talal Arif Shah

NUST Serial no: 281394

____

Signature of Supervisor

Prof. Dr Haider Abbas MCS, NUST

Abstract

Fake or manipulated images propagated through the Web and social media can deceive, emotionally distress, and influence public opinion. The dependability of visual information on the web and the authenticity of digital media appearing virally on social media platforms have raised unprecedented concerns.

The technique of deep fake involves generating human images using neural network tools such as GAN or Auto Encoders. These tools utilize deep learning techniques to overlay target images onto source videos and create videos that appear very realistic. In fact, it becomes nearly impossible to distinguish the difference between deep-fake videos and real videos. This project presents a novel deep learning-based method for effectively differentiating AI-generated fake videos from real videos.

Our approach involves utilizing Res-Next Convolution Neural Networks to capture features at the frame level. These features are then used to train a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) to determine whether a video has been manipulated, i.e., whether it is a deep fake or a genuine video. To improve the real-time performance of the deep fake detection model, we trained it using a combination of different datasets. We utilized videos from Face-Forensic++, Deepfake Detection Challenge, and Celeb-DF datasets to allow our model to learn various features from different types of images. In addition, we tested our model against YouTube videos to achieve competitive results in real-world scenarios.

The proposed methods are evaluated using a dataset of digital forgeries that include 6000 videos from different datasets. For detection, we use 20 frames per video and produce an accuracy of 87.79%.

Table of Contents

List of Figures

List of Tables

Chapter 1: Technical Keywords

1.1 Area of Project

Our project involves the use of deep learning, a subfield of artificial intelligence that utilizes neural networks inspired by the human brain. Computer vision is a crucial aspect of our project, as it allows us to process videos and their frames using OpenCV. To classify whether a video is a deepfake or not, we utilize a PyTorch trained model as a classifier.

1.2 Technical Keywords

- Deep learning
- Res-Next Convolution Neural Network
- Long Short-Term Memory
- Face Recognition
- GAN (Generative Adversarial Network)
- Computer Vision
- PyTorch
- Digital Forensic
- Cloud Computing
- Deepfake Detection

Chapter 2: Introduction

2.1 Project Idea

The emergence of deepfake videos in recent years has made image fabrication a real danger. A deepfake video uses deep learning technology by replacing a person's face, mood, or speech with that of another person with a different emotion or speech. Due to the significant advancements in deep learning-based face alteration techniques over the past few years, it has become more and more difficult to discriminate between actual faces and their visually realistic artificial equivalents. The sophistication of these videos makes it challenging to spot signs of tampering. They may have a significant social, political, and emotional influence on people individually and on society as a whole. There are several scenarios when these realistic face swap deepfakes are used to create political unrest, produce revenge porn, or harass individuals. Nude videos of Brad Pitt and Angelina Jolie are a couple of instances.

The creation of deepfake videos often involves the use of Generative Adversarial Networks (GANs), which consist of two neural networks - a generator and a discriminator. The generator produces synthetic images, and the discriminator distinguishes between the real and fake ones. Through repeated learning and refinement, the generator tries to create images that can deceive the discriminator into identifying them as real. This process continues until the generator produces deepfakes that can successfully fool the discriminator. Some of the popular tools to create deepfakes are FaceApp[1] and FaceSwap[2], which uses pre-trained GAN. Another open source and sophisticated way to create deepfakes id DeepFaceLab GitHub repository [3][4].

We are fighting misinformation and tempered multimedia created by AI by using AI deep learning technology. By using AI, we are combating deepfakes, we have created effective solution to detect and mitigate the spread of these harmful media files. While there are still challenges to be addressed, the development of AI-based solutions for detecting deepfakes offers a promising path forward for combating this growing threat.

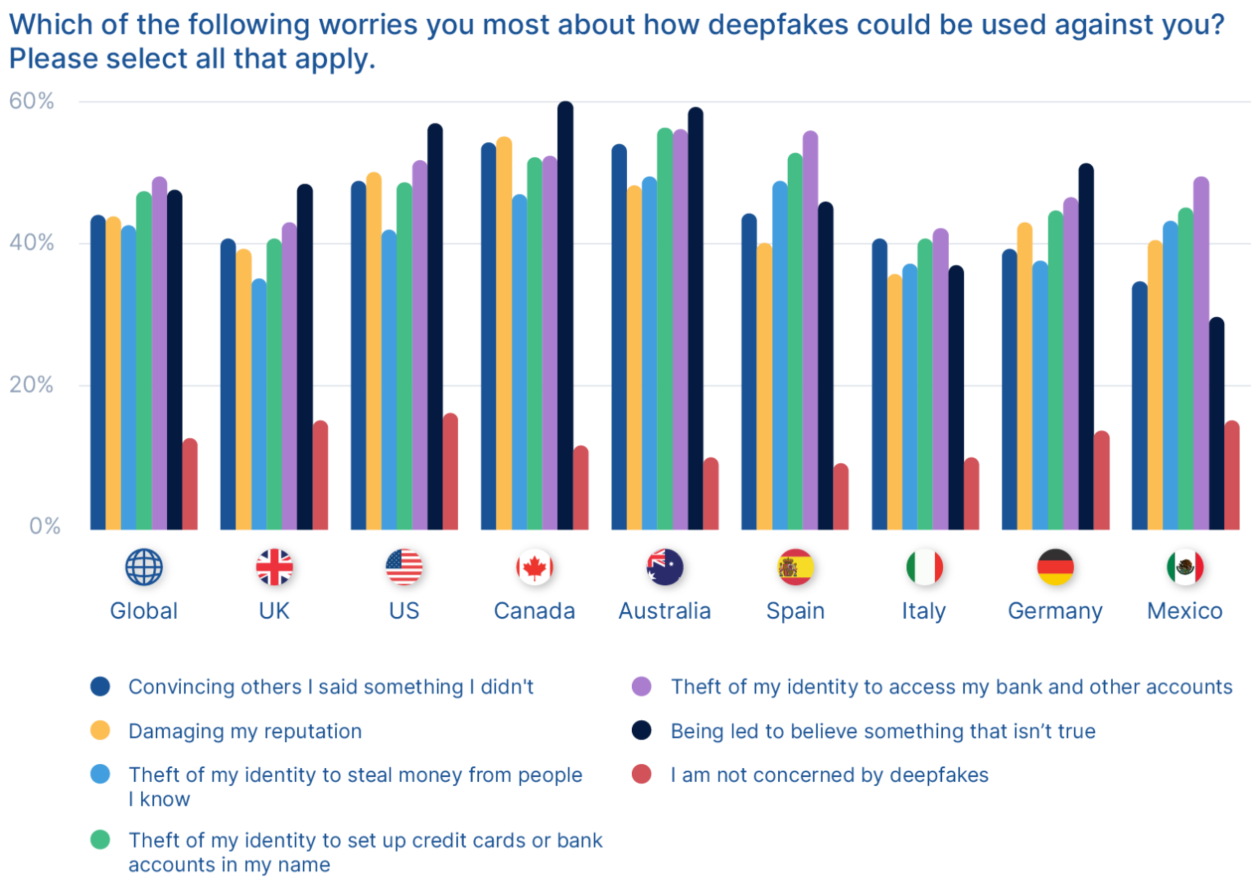


Figure 1: Iproove: How deepfakes can be used against you.

2.2 Motivation of the Project

With the advancement in mobile camera technology and the widespread use of social media and media sharing platforms, the creation and sharing of digital videos has become more convenient than ever before. This is where deep learning technology comes in. It has revolutionized the creation of impossible-to-create technologies only a few years ago. One of these technologies is generative models which can produce incredibly realistic images, speech, music, and even video. The uses of these models are numerous, ranging from making the world more accessible through text-to-speech, to generating training data for medical imaging.

The advancements in technology have presented novel challenges for society to overcome. Deep learning, for instance, has enabled the creation of deep fakes using generative models that can manipulate audio and video clips. Open-source tools and methods for generating deep fakes have become more readily available since their first appearance in 2017, leading to an increase in the number of synthesized media clips. While some deep fakes may be intended for harmless entertainment, others may have harmful consequences for individuals and society. The availability of editing tools and high demand for domain expertise have contributed to the rise in fake videos and their degree of realism.

The proliferation of deep fakes on social media platforms has become commonplace, resulting in spamming and the dissemination of false information across the platform. The potential consequences of a deep fake involving our prime minister declaring war against neighboring countries, or a reputable celebrity verbally abusing their fans, for instance, would be devastating, and could pose a threat to the public by providing misleading and harmful information.

To prevent the harmful effects of deep fakes on social media platforms, it is crucial to detect and identify them. Therefore, a new method based on deep learning has been developed to effectively distinguish fake videos created by AI (Deep Fake Videos) from authentic videos. This technology is essential for identifying and preventing deep fakes from being spread over the internet, as it is crucial to have the ability to recognize and stop their propagation to protect individuals and society from the potential negative consequences.

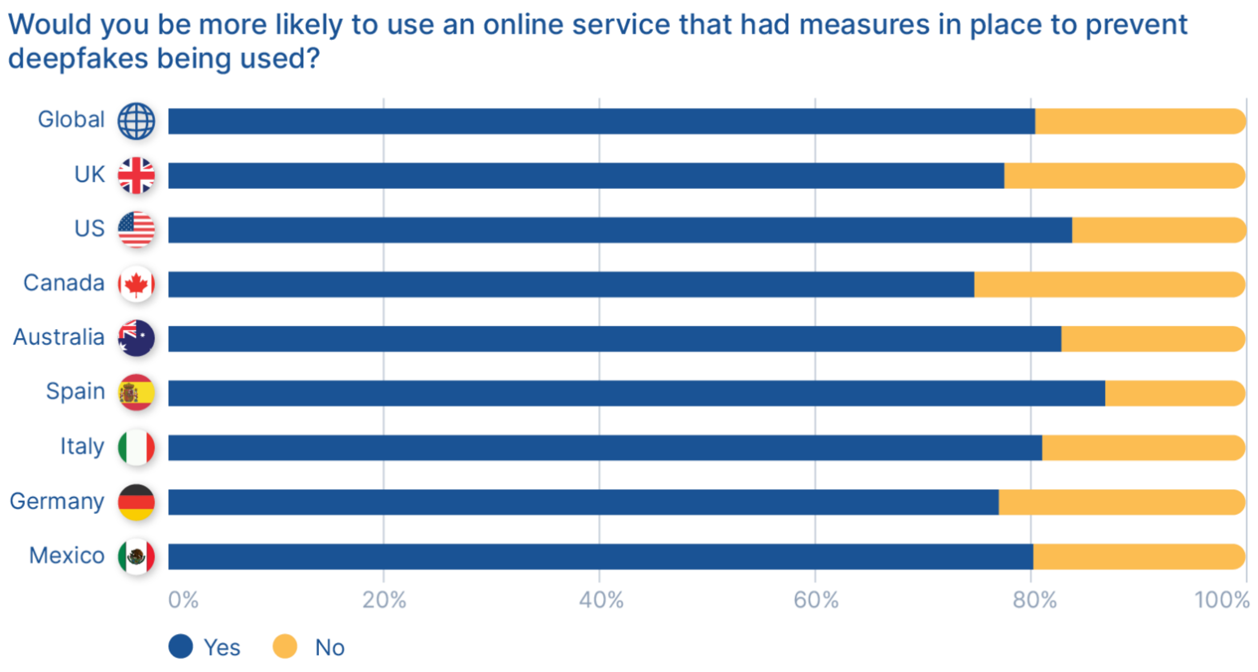


Figure 2: Iproove: Will you like to use online service to prevent deepfakes

2.3 Objectives

2.3.1 General Objectives

- Develop a project to combat deepfakes through the use of machine learning techniques.
- Investigate and expose the distorted truth of deepfakes to reduce abuse and misinformation on the internet.
- Create a system that distinguishes and classifies videos as deepfakes or authentic, using advanced algorithms and computer vision technology.
- Implement an easy-to-use interface for users to upload videos and determine their authenticity.
- Develop and train deep learning models for detecting and identifying deepfake videos.
- Explore and implement different features and techniques for improving the accuracy of deepfake detection.
- Develop and implement practical applications for deepfake detection in media forensics, including integration with social media platforms and other media-sharing websites.
- Educate the public about the dangers of deepfakes and the importance of media literacy and critical thinking when consuming digital content.

2.3.2 Academic Objective

- Developing and implementing policies and procedures for the collection, analysis, and presentation of digital media evidence.
- Ensuring that digital media evidence is presented in a manner that is clear, concise, and understandable.
- Expanding training and education opportunities for digital media forensic investigators.
- Supporting research and development in digital media forensics.
- Encouraging the exchange of information and best practices among digital media forensic investigators and researchers.

- Fostering international cooperation on digital media forensics investigations.
- Supporting the development and use of standards for digital media forensics.
- Promoting the use of digital media forensics in investigations of human rights violations.
- Supporting the use of digital media forensics in investigations of corruption.
- Supporting the use of digital media forensics in investigations of terrorism.

2.4 Sustainable Development Goals

SDG 09: Industry Innovation and Infrastructure

Notable targets

- Encourage the domestic development of technology, research, and innovation in developing nations, particularly by maintaining a policy climate that is favourable to, among other things, the diversification of industries and the addition of value to commodities.
- Efforts should be made to give universal and cheap internet connection in the world's least developed countries by the year 2030. A significant increase in access to information and communications technologies should also be pursued.

How we contribute to the goal

- Promoting Innovation by using modern-day technology and ensuring that digital media evidence is collected and analyzed in a manner that is forensically sound and defensible

SDG 10: Reduced Inequalities

Notable targets

- Promote the inclusion of all individuals regardless of their age, sex, disability, race, ethnicity, origin, religion, or economic status in social, economic, and political aspects
- Work towards achieving equal opportunities and reducing inequality in outcomes by eliminating discriminatory policies, laws, and practices

- Promote the creation of appropriate legislation, policies, and actions to eliminate discrimination and ensure equal opportunities for all.

How we contribute to the goal

- Protecting the integrity of people's information should be a priority, whether the poor or the rich and powerful.

SDG 16: Peace Justice and Strong Institution

Notable targets

- Guarantee the availability of information to the public and safeguard basic liberties through domestic laws and global accords.
- Reinforce pertinent domestic organizations, through international collaboration, to enhance competence at every level, particularly in emerging economies, to prevent aggression and fight terrorism and criminal activities.

How we contribute to the goal

- Technology can have a positive impact on transparency and the effectiveness of justice institutions. But, it also poses a threat when information and data are maliciously manipulated.
- Cybercrime is becoming increasingly prevalent as it is low risk with high potential rewards, leading to more organized criminal groups participating in these activities. Lower- and middle-income countries need to be equipped with the necessary skills and knowledge to combat this crime and protect their populations, while also enabling law enforcement to identify and prosecute cybercriminals.

Chapter 3: Problem Definition and Solution

3.1 Problem Definition

The ubiquitous availability of easy-to-use software for editing digital images brought about by rapid technological advances of the 21st century has dramatically decreased the time, cost, effort, and skill required to fabricate convincing visual forgeries. The technology that allows for manipulating or generating realistic images has far outpaced the technological development of methods for detecting fake imagery. Even experts often cannot rely on visual inspection to distinguish authentic digital images from forgeries. With the proliferation of fake news and deepfake videos, it is more important than ever to be able to identify whether multimedia has been doctored or not. The importance of detecting forged multimedia is protecting against content tampered with or created to deceive. This is especially important in the case of image, audio and video content, as it can be used to create false or misleading information. Forged multimedia can also be used to create unauthorized copies of copyrighted material.

Some of the applications of deepfakes are

3.1.1 Blackmailing

The production of deepfakes raises concerns that manipulated video and audio could be used to deceive people, potentially leading to blackmail or other malicious activities. According to a report by the American Congressional Research Service, deepfakes could be exploited to blackmail public officials or those with access to sensitive information, potentially for purposes of espionage or exerting undue influence.[5]

3.1.2 Pornography

Deepfake pornography has become a concerning issue on the internet since it first emerged in 2017, and is particularly prevalent on Reddit [6]. As of 2019, many of these videos feature non-consenting female celebrities, whose likeness has been manipulated. [7] According to a report by Dutch cybersecurity startup Deeptrace, 96% of all deepfakes online are pornographic. [8] In 2018, a Daisy Ridley deepfake gained attention, among others. [9] This trend highlights the need for effective detection and prevention methods to protect individuals from the harmful effects of deepfakes.

3.1.3 Politics

Deepfakes have been used to misrepresent well-known politicians in videos.

- In April 2018, comedian Jordan Peele worked together with Buzzfeed to produce a deepfake video of Barack Obama with Peele's voice. This video was intended to serve as a public service announcement and raise awareness about deepfakes.[10]
- In the 2020 US Presidential campaign, several deepfakes emerged depicting Joe Biden as having cognitive decline. These deepfakes included videos of him falling asleep during an interview, getting lost, and misspeaking, which were spread to fuel rumors of his decline [11]
- In 2022, Volodymyr Zelensky appears behind a podium, telling Ukrainians to put down their weapons. His head appears too large for and more pixelated than his body - and his voice sounds deeper.[12]
- Gabon’s President, Ali Bongo's video caused coup in the country 2019.[13]
- On September 29, 2020, an advocacy group called RepresentUs uploaded deepfake videos of North Korean leader Kim Jong-un and Russian President Vladimir Putin to YouTube. The purpose of these videos was to serve as public service announcements, warning Americans about the potential interference in US elections by foreign leaders. The deepfakes were intended to be aired as commercials to emphasize the importance of protecting the democracy of the United States. One of the deepfakes featured Vladimir Putin warning Americans about the dangers of election interference and the increasing political divide in the country. [14]

3.2 Statement of Scope

Our deepfake video detection project aims to develop an accurate and efficient deep learning-based system that can effectively detect and classify deepfake videos from real videos. The scope of our project includes developing and training deep neural networks on large datasets of both deepfake and pristine videos, exploring different deep learning architectures and techniques, and implementing the final system as an easy-to-use software application. The ultimate goal of this project is to provide a reliable tool for combating the spread of deepfake videos and preventing their potentially harmful impact on individuals and society.

3.3 Proposed Solution

The emergence of deep learning-based techniques and the reduction in computing costs have made it possible to create realistic videos of human faces, which are commonly referred to as DeepFakes. However, the availability of open-source tools to create DeepFakes has become a major concern as it poses a threat to the authenticity of online media.

To address this, we have developed an open-source online platform called **Media Forensic**, that integrates advanced DeepFake detection methods and provides a user-friendly interface. This platform aims to provide a reliable solution to the detection of DeepFakes and mitigate their potential negative impact on society.

Our system utilizes a Res-Next Convolutional neural network for extracting frame-level features, which are then used to train a Long Short Term Memory (LSTM) based Recurrent Neural Network (RNN) to accurately classify videos as either real or manipulated (i.e. deep fake). To ensure our model performs well in real-time scenarios, we evaluate it on a large, balanced and mixed dataset, created by combining various available datasets like **Face-Forensic++** [15]**, Deepfake Detection Challenge** [16], **and Celeb-DF** [17]. Our results demonstrate that our system achieves competitive accuracy using a simple and robust approach.

3.4 Deliverables

Web Application, where anyone can upload multimedia to identify whether multimedia has been doctored or not.

Chapter 4: Literature Review

4.1 Introduction

Deepfake technology has been rapidly evolving in recent years, allowing for the creation of convincing but fake audiovisual content that can be difficult to detect. This technology poses a significant threat to the integrity of media and information, as deepfakes can be used to spread disinformation, defame individuals, and manipulate public opinion. As such, the development of effective deepfake detection methods has become an important area of research.

Deepfake detection techniques can be broadly classified into two categories: traditional techniques and deep learning-based techniques. Traditional techniques often rely on forensic analysis of video content, such as analyzing inconsistencies in lighting or shadows. However, these methods can be labor-intensive and are limited by the quality of the input video. On the other hand, deep learning-based techniques leverage the power of deep neural networks to learn from large amounts of data and classify videos as either real or fake.

This literature review will provide an overview of recent research in deepfake detection, focusing on deep learning-based techniques. We will examine various approaches to training deep neural networks for detecting deepfakes, including the use of convolutional neural networks (CNNs) and recurrent neural networks (RNNs). We will also analyze the effectiveness of different deepfake datasets for training and testing deepfake detection models. Finally, we will discuss the challenges and future directions of deepfake detection research, including the need for more robust and interpretable models.

4.2 Existing Methods / Research papers

The Face Warping Artifacts method, as described in [18], utilizes a Convolutional Neural Network (CNN) model to detect artifacts by comparing the generated face areas and their surrounding regions. This approach was designed specifically for deepfake videos and targets two types of Face Artifacts. The authors observed that current deepfake algorithms are limited in their ability to generate high-resolution images, which then require further transformation to match the faces in the source video. However, their approach does not incorporate temporal analysis of the video frames, focusing instead on analyzing individual frames.

The Capsule Networks technique proposed in [19] employs a capsule network to identify manipulated images and videos in various scenarios, including detecting replay attacks and computer-generated videos. The method utilizes random noise during the training phase, which is considered less than optimal. However, the approach achieved promising results on their dataset. Nonetheless, it may not be robust enough for real-time data due to the noise introduced during training. Our proposed method aims to train on noiseless datasets and real-time scenarios to achieve improved accuracy and robustness in detecting deepfake videos.

Neural network-based method to detect fake videos. Applied a key video frame extraction technique to reduce the computation in detecting deepfake videos. A model, consisting of a convolutional neural network (CNN) and a classifier network, is proposed along with the algorithm. Xception net is used as CNN model. [20]

The Recurrent Neural Network (RNN) method for deepfake detection, presented in [21], employs sequential processing of frames using an ImageNet pre-trained model. The authors used the HOHO dataset [22], which consisted of only 600 videos. However, the limited size and homogeneity of the dataset may not produce optimal results on real-world data. In contrast, our proposed approach trains the model on a large number of real-time datasets to improve performance on more diverse and realistic data.

Studied the ensemble of different trained Convolutional Neural Network (CNN) models. In the proposed solution, different models are obtained starting from a base network (i.e., EfficientNetB4) making use of two different concepts: (i) attention layers; (ii) Siamese training.[23]

4.3 Startups tackling deepfakes

Detecting deepfake videos has become an increasingly important area of research due to the rising concerns around the potential for mass-disinformation and its impact on society. To address this issue, various deepfake detection tools have been developed with varying degrees of sophistication, accuracy rates, and accessibility to the general public. These tools aim to provide reliable detection techniques to both experts and regular people, allowing them to identify deepfake threats and take necessary actions. As the threat of deepfakes continues to grow, the development of effective and accessible detection tools will become increasingly important for ensuring the integrity of media and maintaining public trust in information sources.

Some of the startups and tools are listed below:

4.3.1 Sensity

Sensity, a Dutch startup, has developed a visual threat intelligence platform and API to detect and combat deepfakes. The platform collects visual threat intelligence from various sources on the open and dark web, using deep learning algorithms to identify malicious visual media and provide a detailed overview of the risks associated with audio-visual content aimed at individuals and businesses. Sensity's detectors leverage AI techniques to detect media manipulation and synthesis, including fake human faces on social media profiles and realistic face swaps in videos. By providing an effective tool to recognize and counter deepfakes, Sensity aims to mitigate the threat of disinformation campaigns and protect the integrity of visual media. [24]

4.3.2 Quantum Integrity

Quantum Integrity, a Swiss startup, has developed a deep learning-based solution for detecting deepfake image and video forgery. Their patented algorithms can be customized for different use cases and are capable of detecting fakes created with various software and techniques. This solution has the potential to save time and money for companies while reducing instances of fraud, and streamlining decision-making processes. It can be applied in areas such as accident reporting, detecting forged documents, and identifying fake videos. [25]

4.3.3 Deepware Scanner

The open-source forensic tool Deepware Scanner, developed by Deepware, specializes in detecting deepfakes and has been actively researching the area since 2018. The tool sets itself apart by conducting deliberate testing on a variety of data sources, including both organic and live videos. Deepware Scanner employs EfficientNet-B7, a convolutional neural network architecture model that uniformly scales all CNN dimensions to enhance accuracy and cost-efficiency. The tool's primary training dataset is CFDC, which contains 120,000 consented videos, and it has been tested on datasets such as 4chan Real, MrDeepFakes, Celeb-DF YouTube, among other. [26]

4.3.4 Sentinel

Sentinel is an Estonian startup that offers solutions to government and media organizations to counter fake media content. The company provides an automated tool for authenticating digital media content by checking if it has been generated using artificial intelligence. Their detection model is based on the Defense in Depth (DiD) approach, which involves a multi-layer defense system comprising a large database of deepfakes and neural network classifiers. This approach allows Sentinel to accurately detect deepfakes and protect their clients against the harmful effects of disinformation. [27]

4.3.5 Deepfake – O – Meter

Deepfake-o-meter is a web-based platform designed for detecting deepfakes in video files. The platform offers several features, including the ability to analyze suspicious video files, run individual algorithms on different servers, and compare the effectiveness of various algorithms on a single input. Users can upload a video via a URL link or as a file, with a maximum size of 50 MB. The platform employs various techniques such as Xception, ClassNSeg, EfficientNet-B3, CNN Detection, spatial pyramid pooling, and mesoscopic image properties analysis to detect fake media. [28]

Chapter 5: Project Methodology

5.1 Methodology

5.1.1 Analysis

After carefully analyzing the problem statement, we evaluated the feasibility of potential solutions. We reviewed various research papers, as mentioned in Section 4.2, and determined that the next step would be to gather and analyze a suitable dataset. During our analysis, we explored different training approaches, such as training the model with only fake or real videos. However, we discovered that this could introduce bias and inaccuracies into the model. After conducting extensive research, we concluded that a balanced training approach would be the most effective way to avoid bias and variance in the algorithm and achieve a high level of accuracy.

Facial parameters identified:

1. Blinking of eyes

2. Teeth enchantment

3. Bigger distance for eyes

4. Moustaches

5. Double edges, eyes, ears, nose

6. Iris segmentation

7. Wrinkles on face

8. Inconsistent head pose

9. Face angle

10. Skin tone

11. Facial Expressions

12. Lighting

13. Different Pose

14. Double chins

15. Hairstyle

16. Higher cheek bones

5.1.2 Design

After conducting research and analyzing the problem statement and dataset, we proceeded to develop the system architecture of the solution. The baseline architecture of the model was decided, which includes various layers and their numbers. We considered several deep learning models like CNN, RNN, Capsule Networks, and others for deepfake detection. After weighing their pros and cons, we decided to use a combination of CNN, RNN and LSTM as our model architecture for the solution.

Our model architecture consists of two main parts, the feature extractor and the classifier. The feature extractor is a convolutional neural network that processes the frames of a video and extracts meaningful features from them. The classifier, which is a recurrent neural network, processes the features extracted by the feature extractor and makes a prediction on whether the video is real or fake. We also added additional layers like batch normalization and dropout to improve the performance of our model. Overall, our proposed model architecture is optimized for deepfake detection and can achieve high accuracy rates.

5.1.3 Development

After careful consideration, PyTorch was selected as the framework for our deepfake detection system due to its compatibility with CUDA and its high degree of customization. CUDA is an API developed by NVIDIA that allows developers to harness the power of GPUs for parallel computing. This means that the PyTorch framework can take advantage of GPUs for faster computation and better performance. Additionally, PyTorch's customization features allow for flexibility in developing and modifying the neural network architecture to meet specific needs.

To train our final model on a large dataset, we decided to use Google Colab Platform. This cloud computing service offers a wide range of powerful tools and services for training, deploying, and scaling machine learning models. By using this platform, we can train our deepfake detection model on a large dataset, improve the accuracy of the model.

5.1.4 Evaluation

After training our model on a large dataset, we evaluated its performance using a confusion matrix. The confusion matrix helps us visualize the performance of the model in terms of true positive, true negative, false positive, and false negative predictions. This approach provides a detailed breakdown of the model's performance, allowing us to identify any weaknesses and potential areas of improvement. By evaluating our model with real-time data, we were able to ensure that it was capable of accurately detecting deepfakes in real-world scenarios.

The use of a confusion matrix also allows us to calculate various performance metrics such as precision, recall, and F1-score. These metrics provide a quantitative measure of the model's performance and help us compare its performance with other deepfake detection models. Additionally, we used cross-validation techniques to ensure the reliability and robustness of our model. By thoroughly evaluating our model's performance, we were able to ensure that it was accurate, reliable, and effective in detecting deepfakes. We evaluated our model with a large number of real time videos which also include YouTube videos.

5.2 Outcome

The trained deepfake detection models are a significant outcome of the solution, as they can assist users in identifying whether a video is real or deepfake. These models can play a crucial role in combating the spread of disinformation and fake news, which has become a significant problem in recent times. By using the models to detect deepfakes, users can prevent the spread of misinformation, protect their reputation, and avoid falling victim to deepfake scams.

They can help to preserve the integrity of media content and promote transparency and trustworthiness in digital media. The models can also be utilized by media organizations, law enforcement agencies, and other institutions to prevent the spread of fake news and malicious propaganda. By using advanced technology to combat the threat of deepfakes, we can work towards building a more informed and trustworthy society.

5.3 Application

The web-based application will have a user-friendly interface, allowing users to easily upload their video files for analysis. Once the video is uploaded, the model will perform pre-processing tasks such as data normalization and feature extraction before making a prediction. The user will receive a binary response indicating whether the video is classified as a deepfake or a real video. Additionally, the application will provide a confidence score for the prediction, which can be used to assess the reliability of the model's prediction.

To ensure scalability and reliability of the web application, it will be hosted on a cloud platform Amazon Web Services (AWS). The application will be deployed using containerization technology using Docker, which allows for easy deployment and scaling. Furthermore, the application will be designed to handle multiple requests simultaneously, ensuring fast and efficient processing of video files. The result of this will be a user-friendly and reliable web-based application for deepfake detection.

5.4 Hardware Resources Required

The processing power required for this project is quite significant, especially when handling large datasets of images and videos. It is essential to have a computer with a high-performance CPU and GPU, as well as sufficient RAM to handle the processing load. The use of GPUs is particularly important as they can accelerate the training of deep learning models by orders of magnitude compared to CPUs alone.

Client Side: Any compatible browser

Serial No.	Parameter	Minimum Requirement
1.	Processor & Generation	Core i5 7th Generation
2.	RAM	16 GB
3.	Hard Disk	100 GB
4.	Graphic Card	NVIDIA GeForce GTX Titan (12 GB RAM)

Table 1: Hardware Requirement

5.5 Software Resources Required

The platform for this project includes an operating system of Windows 10 or 11, a programming language of Python 3.8+, and a framework of PyTorch 1.13 and React. PyTorch is a popular open-source machine learning framework, while React is a high-level web framework that is designed for rapid development and clean, pragmatic design. The cloud platform used for training the model is Google Colab Platform, which provides a scalable and reliable infrastructure for machine learning tasks. The platform also utilizes OpenCV and Face-recognition libraries, which are open-source computer vision and face recognition libraries that are widely used in machine learning applications. To host the web application AWS is used.

Serial No.	**Parameter**	**Requirement**
1.	Operating System	10+
2.	Programming Language	Python
3.	Framework	PyTorch 1.13 & React
4.	Cloud Platform	Google Colab Platform & Amazon Web Services (AWS)
5.	Libraries	OpenCV, Face Recognition

Table 2: Software Requirement

Chapter 6: Project Plan

6.1 Project Task Set

Task 1: Literature Review

The task of literature review is crucial in any research project, as it helps in gaining a deep understanding of the topic at hand and identifying the knowledge gaps in the existing research. In this task, we reviews relevant literature such as academic papers, books, and other sources related to the research topic. The aim is to identify the most relevant and up-to-date sources that will provide valuable insights into the research problem.

Task 2: Data-set gathering and analysis

This involves the crucial step of gathering and analyzing the dataset, which includes downloading the dataset and preparing it for preprocessing. This task requires careful consideration and analysis to ensure that the dataset is appropriate for the deepfake detection model. By properly analyzing the dataset, it can be ensured that it is free from biases, and the deepfake detection model can be trained effectively. This task is essential for building a robust and accurate deepfake detection model.

Task 3: Pre-Processing

The first step is to split the video into frames. This is important as deepfake detection models are trained on individual images rather than videos. After splitting the video into frames, the next step is to crop each frame, keeping only the face region. This is done because deepfakes are usually generated by swapping the face of one person onto the body of another. Therefore, if we focus only on the face region, we can reduce the processing power needed for the model, and also increase the accuracy of the detection process.

Task 4: Implementation of Data Loader

This module includes the implementation of a DataLoader, which is used for loading the video and labels into the model. The DataLoader is implemented to efficiently load the dataset into the model and to preprocess the data for training. This task also involves selecting and implementing an appropriate algorithm for training the model.

Task 5: Hyper parameter tuning

During hyperparameter tuning, the aim is to find the best combination of hyperparameters that can result in the maximum accuracy of the deepfake detection model. Hyperparameters are the parameters that cannot be directly learned from the training data, such as learning rate, batch size, weight decay, and others. By tuning these hyperparameters, the model can be trained more effectively and result in better performance. The tuning process involves evaluating different combinations of hyperparameters on a validation set and selecting the one with the best performance.

Task 6: Training the final model

This task involves training the final model using the best hyperparameters obtained from the previous task. This task requires a large dataset for training, which is usually processed in batches. The data is loaded using a DataLoader, and the model is trained using various optimization techniques such as stochastic gradient descent, Adam optimizer, or any other optimization algorithm. The model is trained to classify between fake and real videos using deep learning techniques. The training process involves the model updating its weights and biases in response to the error rate during the forward and backward propagation steps. Once the model reaches the maximum accuracy, it is saved for deployment in the next task.

Task 7: Front End / User Interface Development

Front-end or user interface development involves designing and developing the visual and interactive components of the application that the end-users will interact with. This includes developing an intuitive user interface design that will enable users to easily upload videos for deepfake analysis and get results.

Task 8: Testing

During the testing phase, the functionality and usability of the application is thoroughly evaluated to ensure that it meets the requirements of the stakeholders. The testing phase is an essential part of the software development life cycle, as it helps to identify and correct any defects or issues in the application before it is released to the end-users. It also helps to ensure that the application is user-friendly and meets the specified requirements.

Task 9: Deployment

The final task involves deploying the model in a production environment where it can be accessed by the end-users. The model is integrated into a web-based application where the user can upload the video and receive a prediction of whether the video is deepfake or real. The application uses a pre-trained model to predict the authenticity of the video. The deployment involves creating a user interface, integrating the model, and hosting the application on a server. The deployment process involves ensuring that the model can handle multiple requests, and the response time is optimized. Once the application is deployed, it is monitored to ensure that it is functioning optimally, and any issues are addressed promptly.

6.2 Project Timeline

Please refer Annex

Chapter 7: Software Requirement Specification

7.1 Introduction

7.1.1 Purpose and Scope of Document

This document provides a comprehensive project plan for the development of a deepfake video detection system using neural network technology. The project is aimed at developers and sponsors who are interested in understanding the functionality of the system, its scope, and the development process. The document outlines the key aspects of the project, including the use case diagram, data flow diagram, activity diagram, functional and non-functional requirements, project risks and mitigation strategies, and metrics and measurements for tracking project progress.

The project will involve the use of PyTorch and Python programming language to develop the deepfake detection system. The system will be deployed on a web-based platform that allows users to upload and submit videos for processing. The system will analyze the video using machine learning algorithms to determine whether it is a deepfake or a real video. The project will follow a well-defined development process that involves tasks such as data-set gathering and analysis, module implementation, hyper parameter tuning, user interface development, testing, and deployment.

The project plan will serve as a roadmap for the successful development and deployment of the deepfake video detection system. It outlines the key milestones, timelines, and deliverables of the project, which will ensure that the project is completed on time, within budget, and to the satisfaction of all stakeholders. By following the plan, the development team can ensure that the project is executed efficiently and effectively, and that the final product meets all the requirements of the end-users.

7.1.2 Use Case View

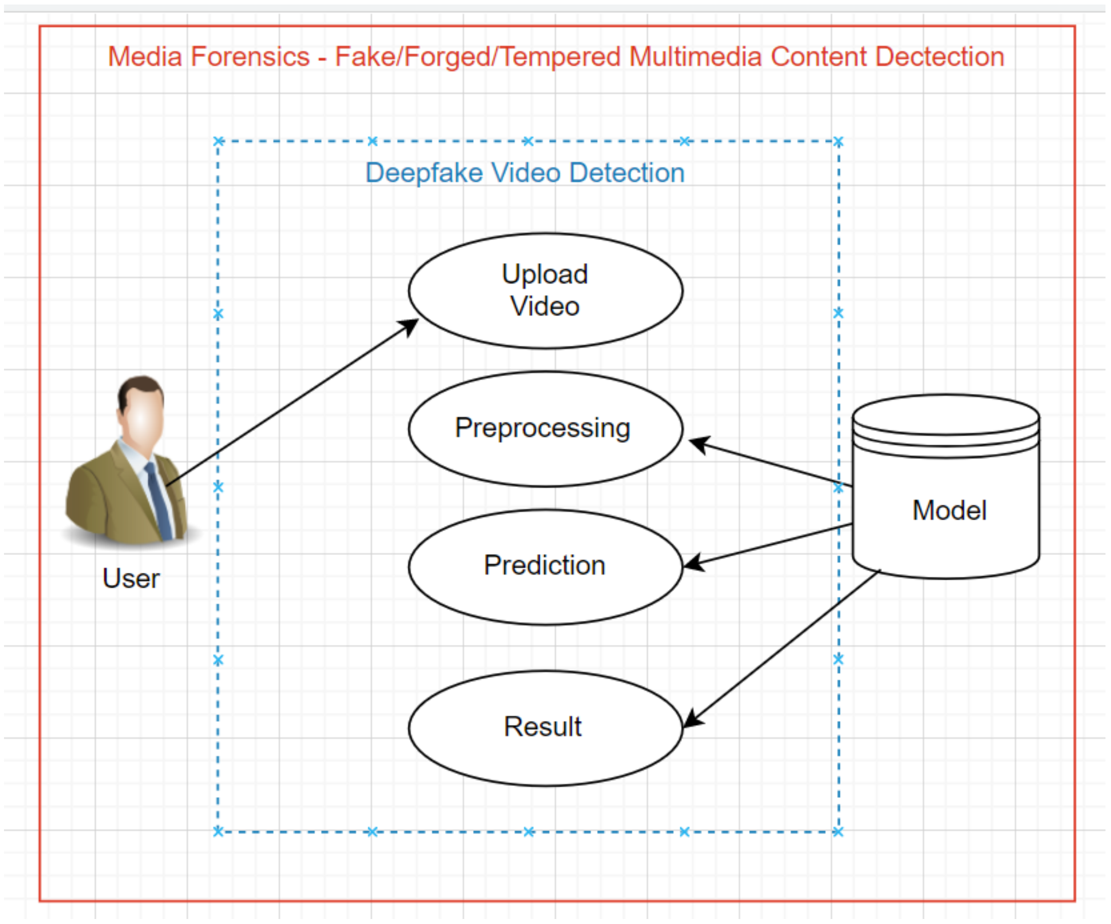


Figure 3: Use Case Diagram

7.2 Functional Model Description

A structured analysis approach involves breaking down a system into its constituent parts and analyzing how these parts interact with one another. The data flow diagram is used to illustrate the flow of data within a system, including how data is input, processed, and output. In the context of the Deepfake video detection system, a data flow diagram could be used to illustrate how video data is received and processed by the system, and how the system outputs a result indicating whether or not the video is a deepfake.

In contrast, an object-oriented approach involves representing a system in terms of objects and their interactions. An Analysis Class diagram can be used to illustrate the structure of a system, including the classes and their relationships to one another. In the context of the Deepfake video detection system, an Analysis Class diagram could be used to illustrate the classes involved in the system, such as the video input class, the pre-processing class, the deepfake detection class, and the output class. The class diagram can also illustrate the methods and attributes of each class and how they interact with each other.

7.2.1 Data Flow Diagram

Data Flow Diagram Level 0

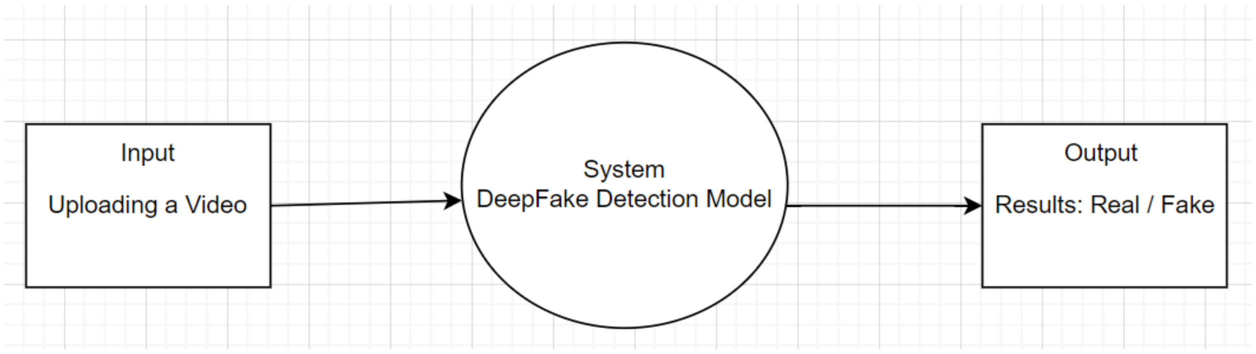


Figure 4: DFD Level 0

The DFD level-0 of the deepfake detection system has been described with the following details:

• Input: The input to the system is the uploaded video which needs to be analyzed for the presence of deepfakes.

• System: The system receives the uploaded video and performs all the necessary processing to analyze it for the presence of deepfakes. It analyzes the frames of the video, crops the faces, and extracts the required features for the deepfake detection model. The system also uses the trained deepfake detection model to predict if the uploaded video is real or fake.

• Output: The output of the system is a binary decision on whether the uploaded video is a deepfake or a real video. The system displays the result to the user along with all the details of the analyzed video.

Overall, the DFD level-0 of the deepfake detection system provides a high-level view of the system's data flow, which helps in understanding the system's behavior. It outlines the input, processing, and output of the system and provides a clear picture of how the system functions.

Data Flow Diagram Level 1

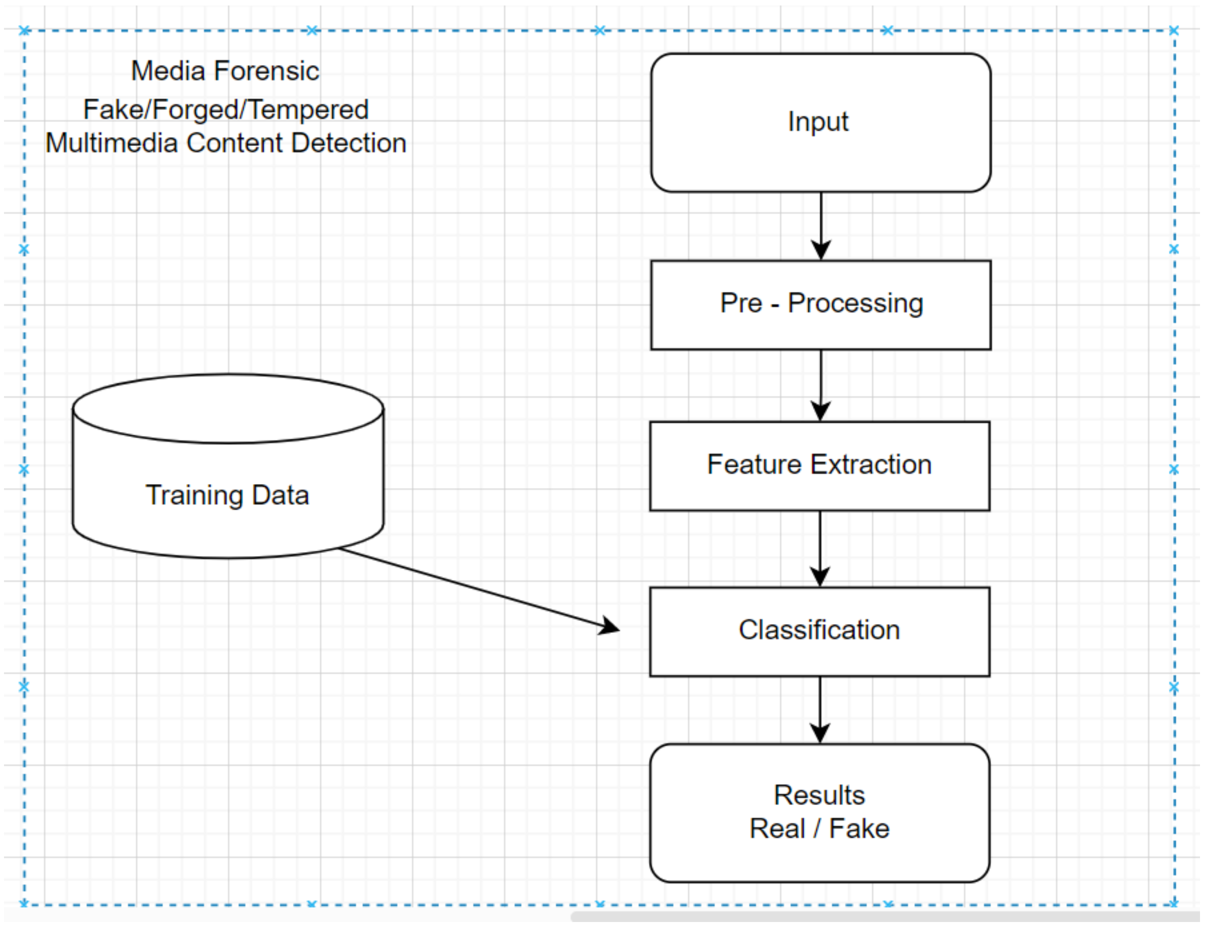


Figure 5: DFD Level 1

In DFD Level – 1, the system's functions are broken down into sub-functions, which provide more detail on how data flows through the system. The sub-functions are presented as boxes, and the data flows are shown as arrows between these boxes. The input and output of each box are also shown in the diagram, allowing for a more detailed view of the system's functionality. By breaking down the system into its sub-functions, DFD Level – 1 provides a more comprehensive understanding of the system and its processes.

Data Flow Diagram Level 2

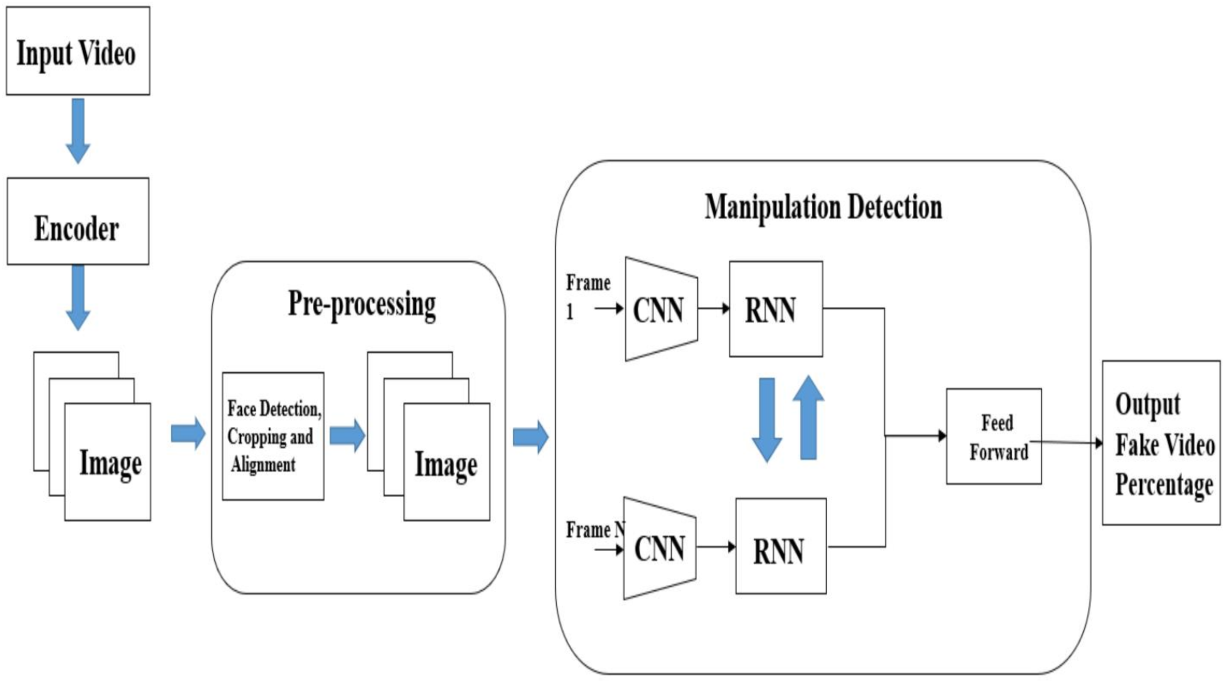


Figure 6: DFD Level 2

For the Deepfake detection project, DFD level 2 can provide a more detailed view of the system by breaking down the sub-processes identified in DFD level 1 into further sub-processes. For example, a sub-process of face detection could be further broken down into sub-processes for facial landmark detection, face alignment, and face cropping. Each sub-process would be depicted as a separate process on the level 2 DFD. The data flows and data stores associated with each sub-process would also be shown, providing a more comprehensive view of the system. DFD level 2 can provide a more granular view of the system's processes and data flows, enabling better understanding and optimization of the system.

7.2.2 Activity Diagram

Training Workflow

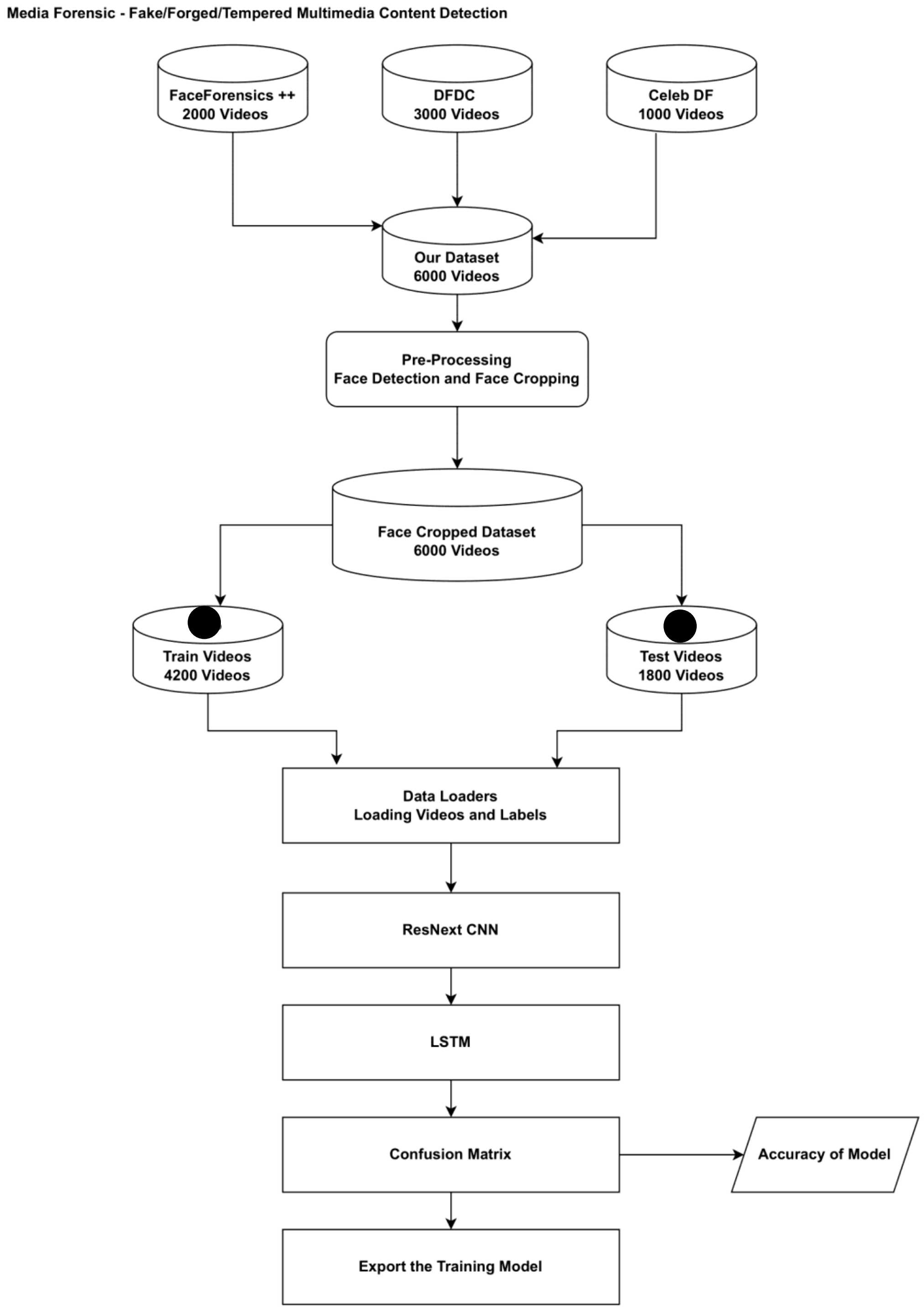

Media Forensic - Fake/Forged/Tempered Multimedia Content Detection
FaceForensics ++ 2000 Videos
DFDC 3000 Videos
Celeb DF 1000 Videos
Our Dataset 6000 Videos
Pre-Processing Face Detection and Face Cropping
Face Cropped Dataset 6000 Videos
Train Videos 4200 Videos
Test Videos 1800 Videos
Data Loaders Loading Videos and Labels
ResNext CNN
LSTM
Confusion Matrix
Accuracy of Model
Export the Training Model

Figure 7: Training Workflow

Testing Workflow

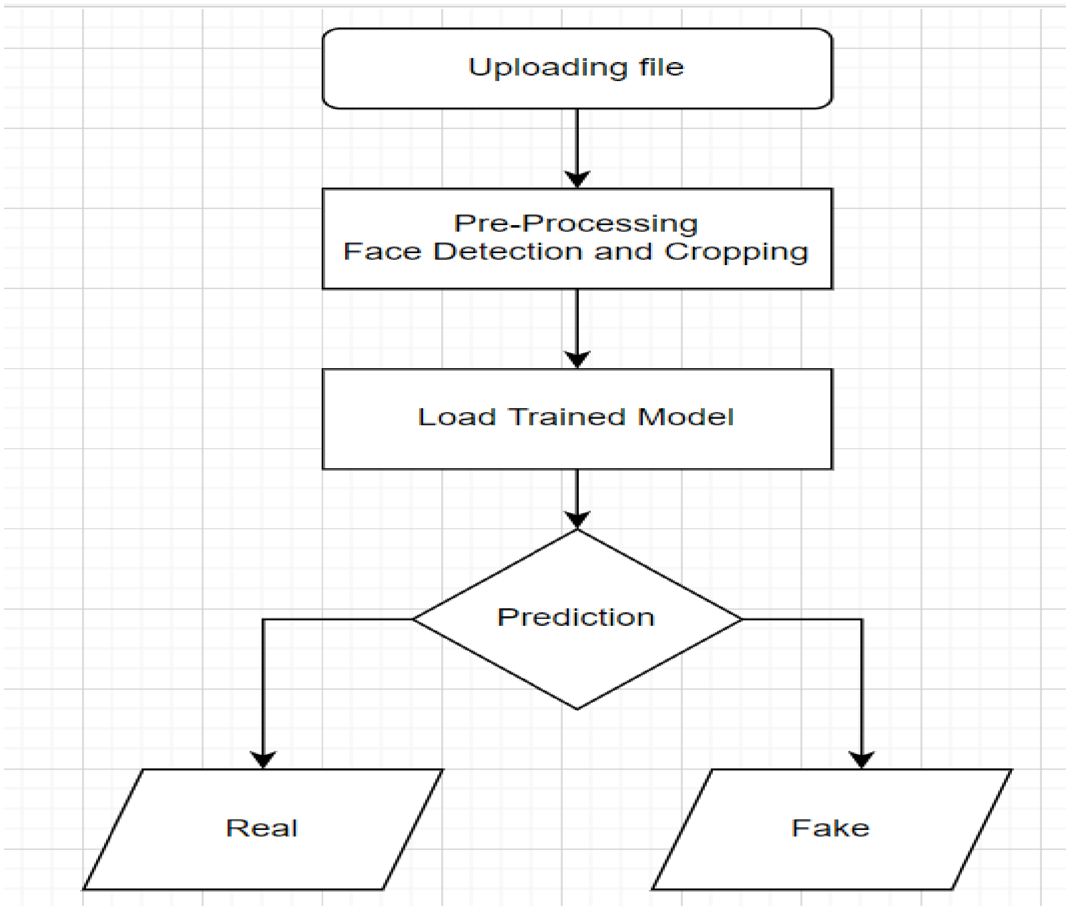


Figure 8: Testing Workflow

7.2.3 Non-Functional Requirement

Performance Requirement

• The system must be able to accurately detect deepfake videos, with a high level of reliability.

• The user interface should be intuitive and user-friendly, with clear instructions for uploading and analyzing videos.

• The system should be optimized for speed and efficiency, so that it can process videos quickly and accurately.

• The system should be adaptable to different video formats and sources, with the ability to analyze videos from a variety of platforms and devices.

• The system should be designed with future upgrades and improvements in mind, with the flexibility to integrate new features and technologies as they become available.

Safety and Security Requirement

• The system implements strong data security measures to prevent any unauthorized access to the user's video and personal information.

• The system ensures the privacy of the users by not storing any data related to the video once the processing is completed.

7.2.4 Sequence Diagram

A sequence diagram is a type of interaction diagram that visualizes the flow of messages and interactions between objects or components of a system. In the context of our project, a sequence diagram can be used to illustrate the flow of information between various components of the system such as the user interface, the neural network model. For instance, the sequence diagram illustrate the steps involved in uploading a video, processing the video using the deepfake detection algorithm, and displaying the result to the user.

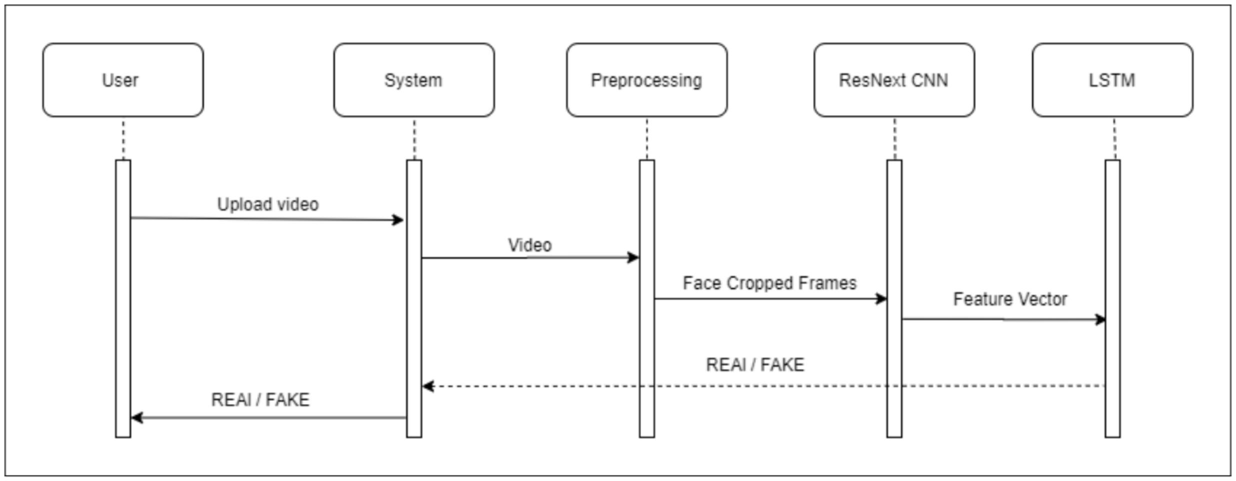


Figure 9: Sequence Diagram

Chapter 8: Software Design Specification

8.1 Introduction

8.1.1 System Architecture

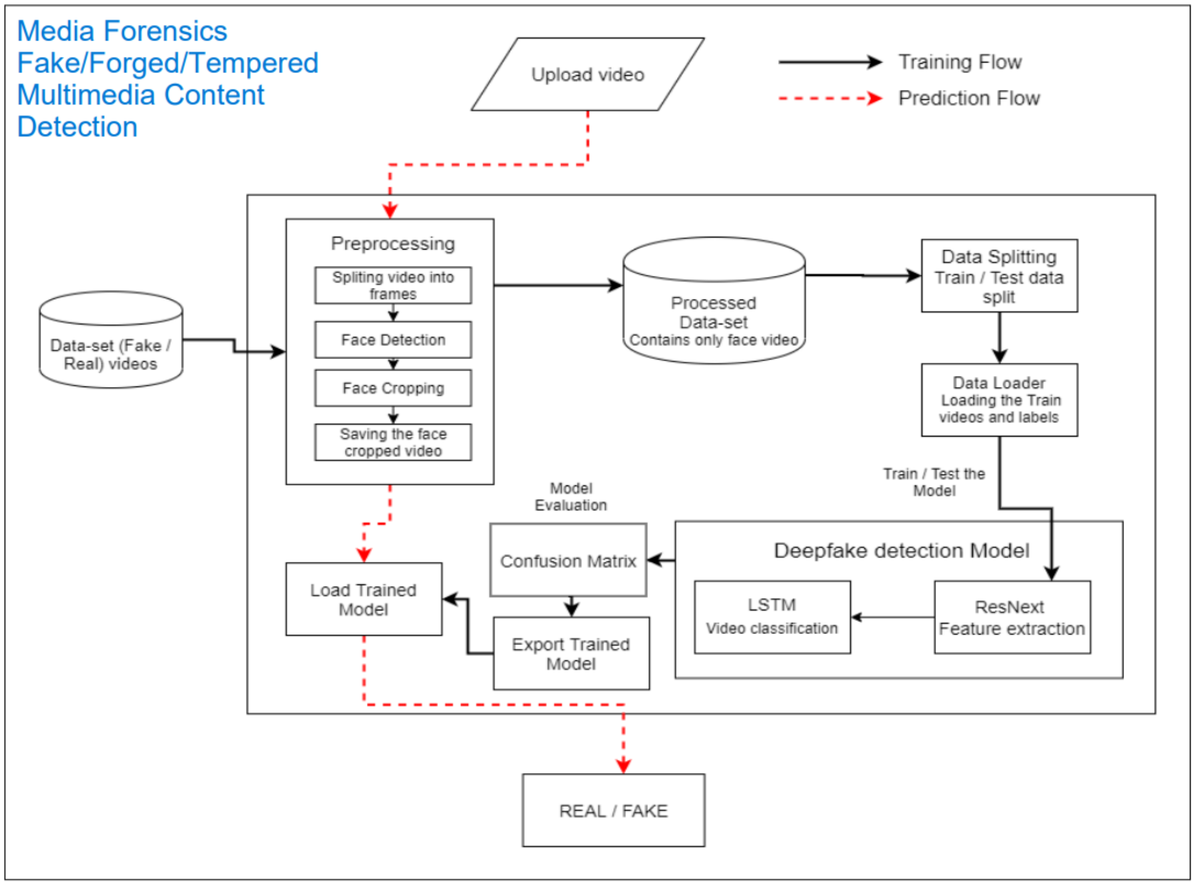


Figure 10: System Architecture

The system architecture of the deepfake detection model is based on a convolutional neural network (CNN). The CNN has multiple layers of convolutional and pooling layers followed by fully connected layers that perform the classification. The model is trained on a dataset of equal numbers of real and fake videos to avoid bias in the model. In the development phase, the dataset is preprocessed and only includes face-cropped videos to improve the performance of the model.

The CNN-based model architecture has shown promising results in detecting deepfake videos with high accuracy. The use of preprocessed face-cropped videos in training the model has helped to improve the performance of the model. By training the model on an equal number of real and fake videos, the model can detect deepfake videos with high accuracy and without any bias.

8.1.2 Deepfakes Creation

To detect deepfake videos, it is crucial to have a clear understanding of the creation process involved in deepfake generation. Most of the tools used for creating deepfakes, such as GANs and autoencoders, take a source image and a target video as input. These tools then identify the face in the video, replace the source face with the target face on each frame, and combine the frames using pre-trained models to create a realistic-looking deepfake. However, these models often leave behind traces or artifacts that are not easily noticeable to the naked eye. The goal of this project is to identify these subtle traces and artifacts that distinguish deepfake videos from real videos.

In this project, the same approach has been used to detect deepfakes. The deepfakes created using pre-trained neural network models are so realistic that they are almost impossible to differentiate from real videos with the naked eye. However, by carefully analyzing the video frames and identifying these distinguishable artifacts, it is possible to distinguish between deepfake and real videos. The project aims to develop an efficient and reliable system for detecting deepfakes and ensuring the authenticity of the videos uploaded to the system. By understanding the intricacies of deepfake creation and developing advanced algorithms, the system will provide an effective tool for detecting deepfakes and promoting the integrity of media content.

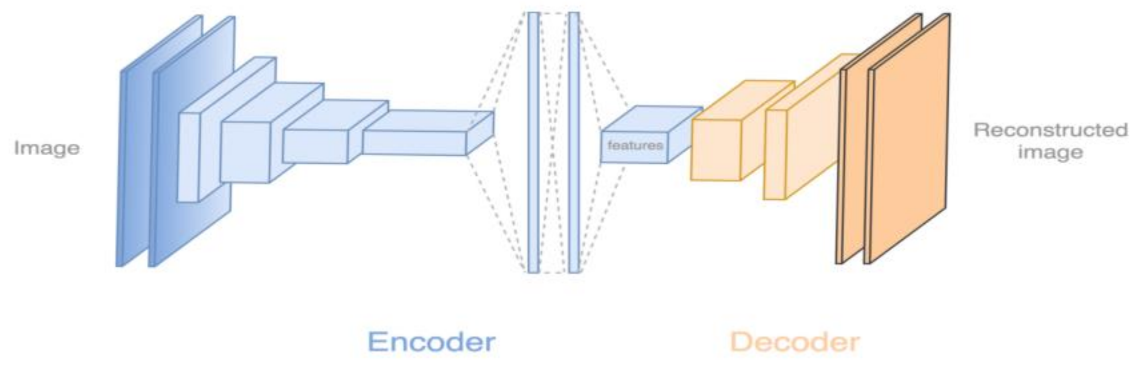


Figure 11: Deepfake generation: Auto Encoders

Generative Adversarial Network GAN

The process of creating a deepfake involves the use of advanced algorithms such as generative adversarial networks (GANs). These algorithms are designed to create fake digital content that is almost indistinguishable from real content. GANs consist of two main components - a generator and a discriminator. The generator creates the fake content, while the discriminator tries to identify whether the content is real or fake. As the discriminator provides feedback to the generator, it learns to improve its output, creating deepfakes that are increasingly difficult to distinguish from real content. As deepfake technology advances, it is important to develop countermeasures that can detect and prevent the spread of this kind of manipulated content. The detection of deepfakes is an ongoing challenge, and researchers are continuously developing new methods to improve the accuracy and reliability of deepfake detection tools. [29]

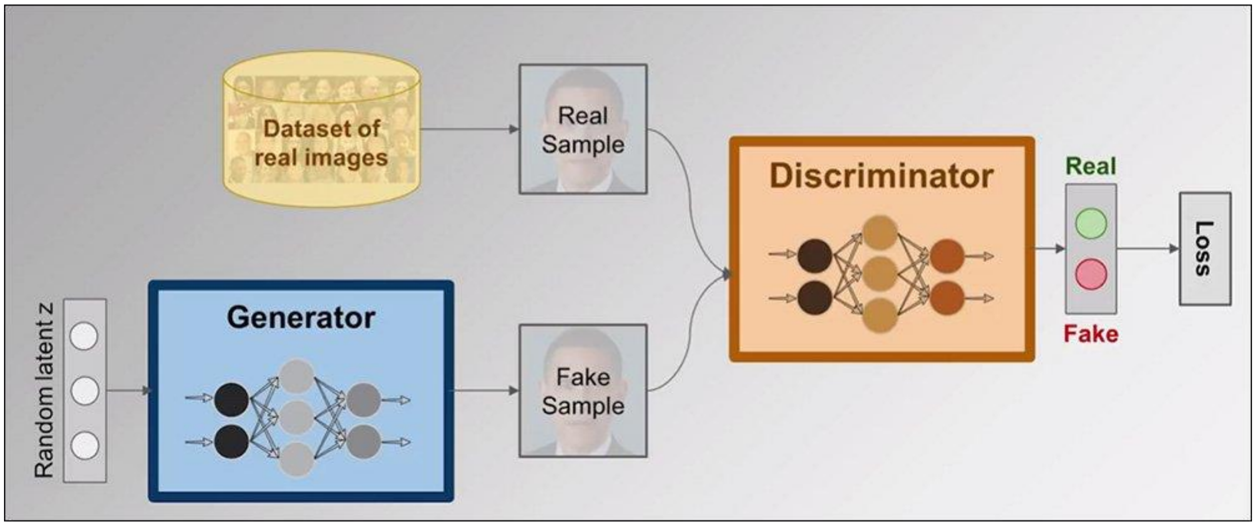


Figure 12: Generative Adversarial Networks [30]

Tools for Deep Fake Creation

1. FaceSwap

2. Faceit

3. Deep Face Lab

4. Deepfake Capsule GAN

5. Large resolution face masked

6. FaceApp

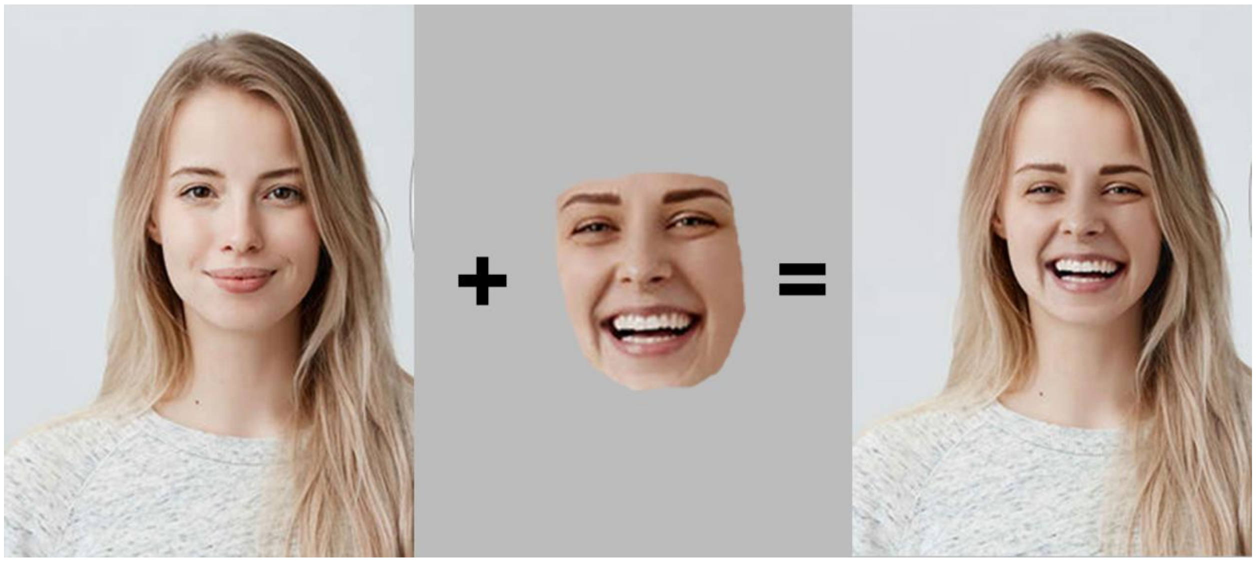


Figure 13: Face Swap [31]

8.2 Architectural Design

8.2.1 Module 1: Dataset Gathering

In order to make the deepfake detection model efficient for real-time prediction, we gathered data from various available datasets such as FaceForensic++ [15], Deepfake Detection Challenge (DFDC) [16], and Celeb-DF [17]. To ensure accurate and real-time detection on different types of videos, the team combined the collected datasets and created a new dataset. To avoid training bias, the team considered 50% real and 50% fake videos in the dataset.

The DFDC dataset consisted of certain audio-alerted videos that were out of the scope for this project. We preprocessed the DFDC dataset and removed the audio-altered videos by running a Python script. After preprocessing the DFDC dataset, we selected 1500 real and 1500 fake videos from the DFDC dataset, 1000 real and 1000 fake videos from the FaceForensic++ dataset, and 500 real and 500 fake videos from the Celeb-DF dataset. This made the total dataset consist of 3000 real, 3000 fake videos, and 6000 videos in total.

To ensure that the deepfake detection model can detect the differences between real and fake videos, the team used the dataset that had a balanced distribution of real and fake videos.

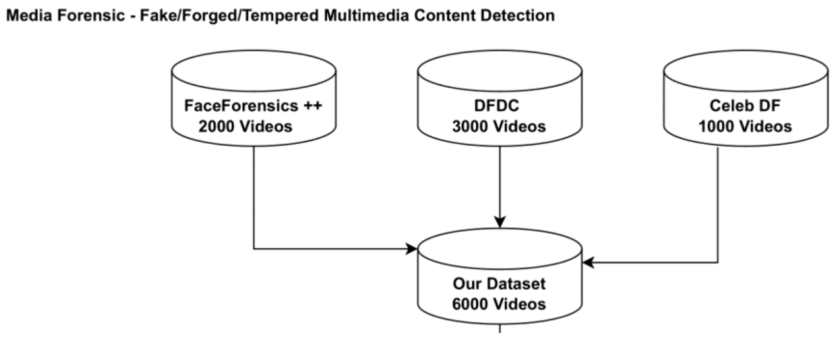


Figure 14: Dataset Gathering

8.2.2 Module 2: Pre-Processing

In Module 2 of project, the focus is on pre-processing the videos to remove unwanted noise and retain only the necessary information, which is the face. The first step in the pre-processing of the videos is to split them into frames. Each frame is then scanned to detect the presence of a face, and the face is then cropped. This process is carried out for each frame of the video. Once all the frames of the video are processed, the cropped frames are combined to create a new video containing only the face of the individual.

To maintain uniformity in the number of frames, a threshold value is selected based on the mean of the total frames count of each video. This threshold value is selected based on the computational power of the system in the experimental environment. The number of frames in a video can be computationally intensive, and it becomes difficult to process all frames at once. As a video of 10 second at 30 frames per second(fps) will have total 300 frames and it is computationally very difficult to process the 300 frames at a single time in the experimental environment. In this project, a threshold value of 150 frames is selected to maintain the computational efficiency of the system.

To ensure that the project uses the Long Short-Term Memory (LSTM) properly, the frames are considered sequentially and not randomly. Only the first 150 frames of the video are saved in the new video, and the newly created video is saved at a frame rate of 30 fps and resolution of 112 x 112. This resolution is selected based on the experiment's computational power and the objective of the project.

By using this pre-processing approach, the system can reduce the noise in the video and retain only the necessary information. This approach allows the model to focus on the essential features and increase the accuracy of the deepfake detection system. Pre-processing is a crucial step in the deepfake detection process, and this module focuses on efficiently and accurately pre-processing the videos to create a new processed dataset.

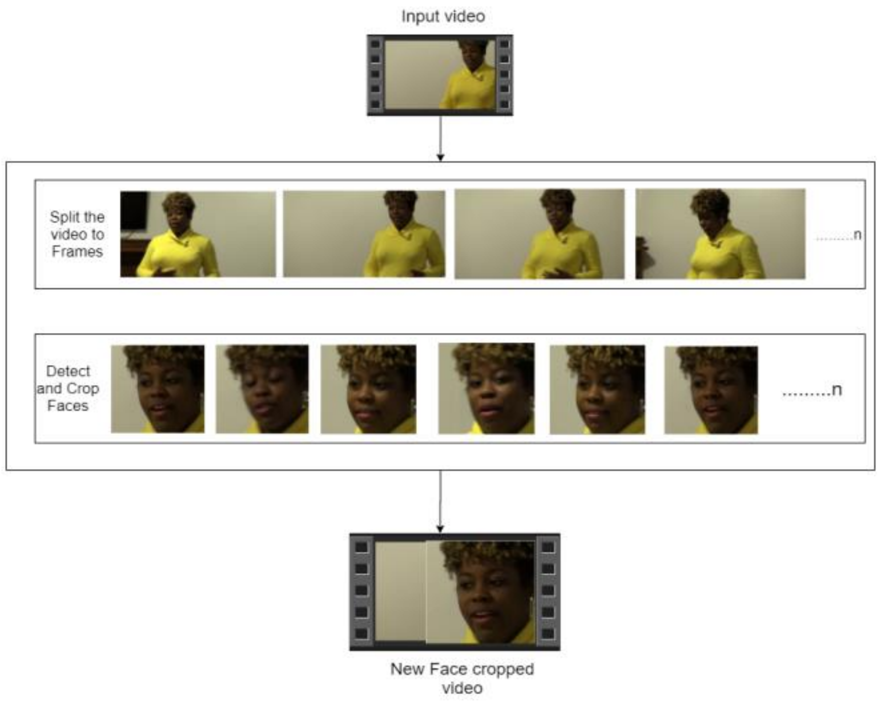


Figure 15: Pre-Processing

8.2.3 Module 3: Dataset Splitting

Module 3 of the deep fake project involves the splitting of the dataset into train and test sets. This step is crucial for evaluating the performance of the deep fake detection model. The dataset is split in a balanced manner, with 50% real videos and 50% fake videos in each split. The ratio of the train and test dataset is set to 70% and 30%, respectively. This ensures that the model is trained on a sufficient amount of data while still having a significant amount of data for testing.

Splitting the dataset into train and test sets is necessary to evaluate the performance of the deep fake detection model. If the model is trained on the entire dataset, it may result in overfitting, which means the model is unable to generalize well to unseen data. By splitting the dataset into train and test sets, the model can learn from the train set and evaluate its performance on the test set. This helps in determining the effectiveness of the model and identifying any areas of improvement.

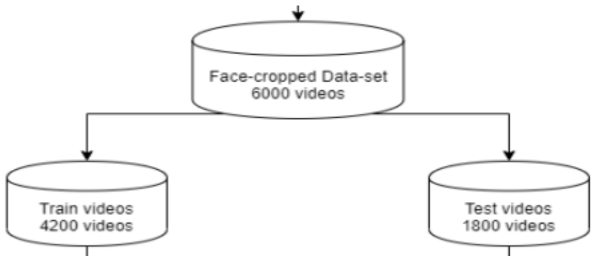


Figure 16: Dataset splitting

8.2.4 Module 4: Model Architecture

ResNext Model Architecture

Our deepfake detection model is a combination of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), and is built upon the pre-trained ResNext model. ResNext is a Residual CNN network optimized for high performance on deeper neural networks. For our experiments, we have used the resnext50_32x4d model, which consists of 50 layers and has 32 x 4 dimensions. We have fine-tuned the pre-trained ResNext model by adding extra layers and selecting a proper learning rate to properly converge the gradient descent of the model. The 2048-dimensional feature vectors after the last pooling layers of ResNext are used as the sequential LSTM input.

LSTM for Sequence Processing

Our model uses a Long Short-Term Memory (LSTM) layer for sequence processing of the video frames. The 2048-dimensional feature vectors are fitted as the input to the LSTM layer. We have used 1 LSTM layer with 2048 latent dimensions and 2048 hidden layers along with 0.4 chance of dropout, which is capable of achieving our objective. The LSTM layer processes the frames in a sequential manner so that the temporal analysis of the video can be made by comparing the frame at ‘t’ second with the frame of ‘t-n’ seconds, where n can be any number of frames before t. The model also consists of a Leaky Relu activation function, a linear layer of 2048 input features and 2 output features, and an adaptive average pooling layer with the output parameter 1. A sequential layer is used for sequential processing of the frames, and a SoftMax layer is used to get the confidence of the model during prediction.

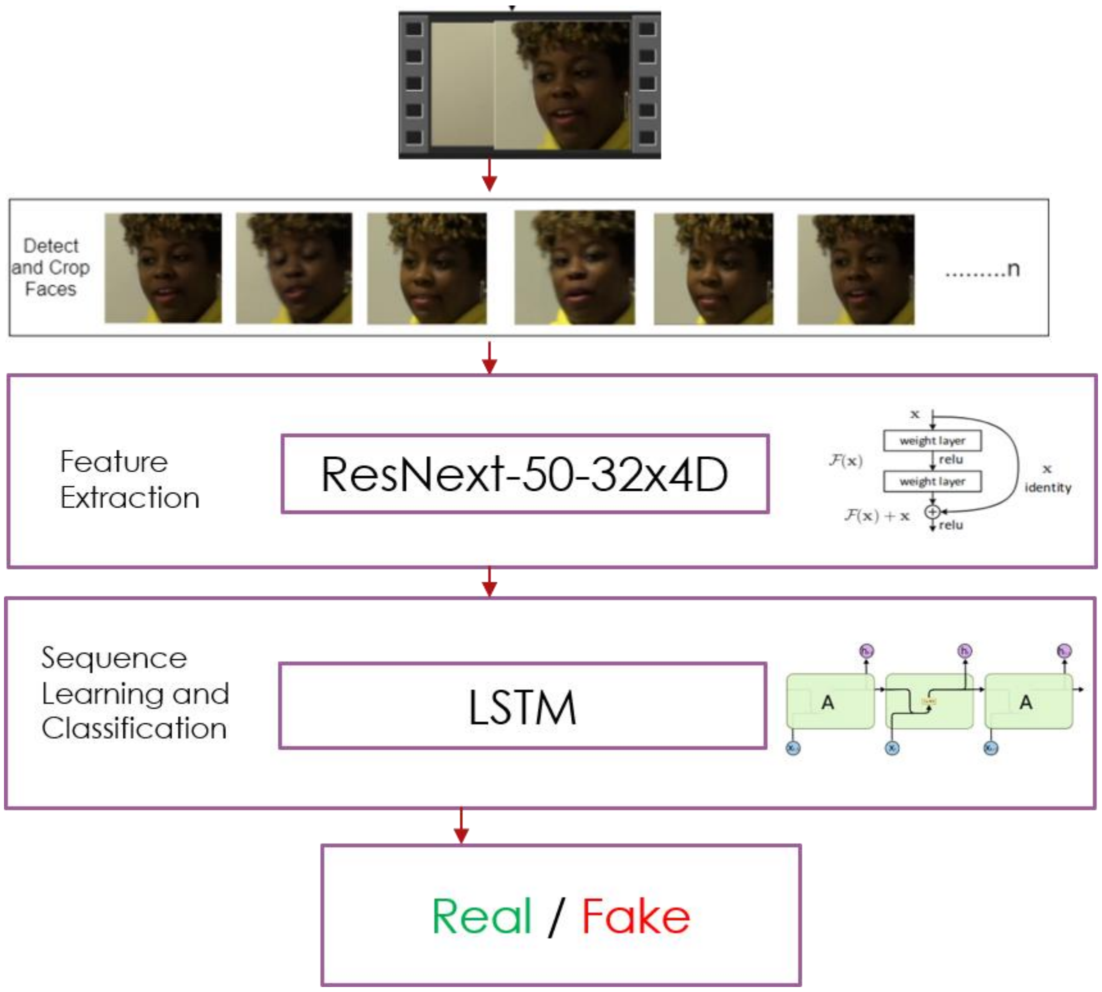


Figure 17: Model Architecture

8.2.5 Module 5: Hyper-Parameter Tunings

Hyper-parameter tuning is a crucial step in deep learning as it determines the performance of the model. After multiple iterations and experiments, the best hyper-parameters for the deep fake detection model have been chosen. For adaptive learning rate, the Adam optimizer [32] has been used with a learning rate of 1e-5 (0.00001) to achieve a better global minimum of gradient descent. The weight decay used is 1e-3. Cross-entropy approach is used for calculating the loss, as this is a classification problem. To optimize the computation power, batch training has been implemented with a batch size of 4, which has been found to be the ideal size for training in the development environment.

The hyper-parameters are selected based on the trade-off between accuracy and computational complexity. Selecting a higher learning rate would speed up the training but may result in overshooting and missing the global minimum. Similarly, a lower learning rate would provide better accuracy but would slow down the training process. The batch size is another crucial hyper-parameter, as it determines the amount of data processed in each iteration. A large batch size would require more memory and computation power but would be faster, while a small batch size would be more accurate but would be slower. The weight decay is used to prevent overfitting, and the cross-entropy loss function is used for multi-class classification. All these hyper-parameters work together to create an optimal deep fake detection model with high accuracy and low computational complexity.

8.2.6 Module 6: User Interface

The user interface is a critical component of any software application, and in the case of our project, it plays a crucial role in providing a seamless experience to the end-users. React framework is used to develop the user interface of the application, which is a popular choice among developers due to its component-based approach and efficient rendering capabilities. The scalability of the application is also taken into consideration while choosing React as the framework since it allows for easy maintenance and modification of the codebase as the project evolves.

The first page of the user interface is designed to be simple and intuitive, with a single button for uploading the video. Once the user uploads the video, it is passed to the model for prediction, and the output is displayed on the screen. The prediction made by the model is whether the video is real or fake, and it is presented to the user in a clear and concise manner.

Chapter 9: Project Implementation

9.1 Introduction

Deepfake technology has become a significant concern for society due to its potential for misuse. There have been numerous instances of deepfake videos being used to deceive people on social media platforms. These videos are often created using the faces of famous personalities, and they create panic among normal people. The need to spot these deepfakes accurately has become essential to distinguish them from real videos.

There are numerous instances where deepfake creation technology is used to deceive users on social media platforms by disseminating fake deepfake videos of well-known individuals, including Mark Zuckerberg from the House A.I. Hearing, Donald Trump from the Breaking Bad series, where he played James McGill, Barack Obama from the PSA, and many others [33]. Deepfakes of this nature causes a great deal of panic in the average person, making it necessary to identify these deep fakes precisely in order to tell them apart from the real footage.

Recent advances in technology have changed the field of video manipulation. Open-source deep learning frameworks such as TensorFlow, Keras, and PyTorch, combined with cheap access to high computation power, have driven a paradigm shift. Autoencoders [34] and Generative Adversarial Network (GAN) pretrained models have made the tampering of realistic videos and images very easy. The proliferation of deepfake creation applications [1][2] that use these models has made it possible for anyone to create highly realistic synthesized transformations of faces in real videos. These apps provide the user with functionalities like changing the face's hair style, gender, age, and other attributes, and allow them to create high-quality and indistinguishable deepfakes.

While some malignant deepfake videos exist, they remain a minority. However, the released tools that generate deepfake videos are being extensively used to create fake celebrity pornographic videos or revenge porn [35]. Some examples include Brad Pitt and Angelina Jolie nude videos. The realistic nature of the deepfake videos makes famous personalities the target of pornographic material, fake surveillance videos, fake news, and malicious hoaxes. Deepfakes are also popular for creating political tension [36]. Detecting deepfake videos and preventing their dissemination on social media platforms is crucial to prevent their harmful effects.

9.2 Tools and Technologies used

9.2.1 Planning

- Jira Software

9.2.2 UML Tools

- Draw.io

9.2.3 Programming Languages

- Python
- JavaScript

9.2.4 Programming Framework

- PyTorch
- React

9.2.5 IDE

- Python IDLE
- Google Colab
- Jupyter Notebook

9.2.6 Version Control

- Git

9.2.7 Cloud Services

- Amazon Web Services (AWS)
- Docker

9.2.8 Application and Web Server

- Amazon Elastic Compute Cloud (Amazon EC2)

9.2.9 Libraries

- Torch
- Torch Vision
- Numpy
- Open CV
- Face-Recognition
- Json
- Pandas
- Sklearn
- Pickle
- Dlib
- Pillow

9.3 Algorithm Details

9.3.1 Dataset Details

Refer to Section 8.2.1

9.3.2 Preprocessing Details

In the preprocessing stage of the deepfake detection project, various steps were taken to prepare the videos for training the model. Firstly, the videos were imported into the project using the glob module and stored in a Python list. Next, the cv2.VideoCapture method was used to read the videos and calculate the mean number of frames in each video. To maintain consistency, a value of 150 was selected as the ideal number of frames for creating the new dataset.

After determining the optimal number of frames, the videos were split into individual frames and the faces within each frame were cropped to focus on the key area for analysis. These face-cropped frames were then written to a new video file using the VideoWriter method. The new video was written in the MP4 format at a frame rate of 30 frames per second with a resolution of 112 x 112 pixels.

To enable the proper use of LSTM for temporal sequence analysis, only the first 150 frames of each video were selected for use in the new dataset. By selecting the initial frames of the videos, the model could analyze the sequence of frames leading up to the prediction, which allowed for a more accurate and detailed assessment of each video.

The pre-processing stage of deepfake detection is crucial for developing an accurate and effective model. By selecting the appropriate frames and focusing on the critical facial features, the model can more accurately detect and classify deepfake videos. By utilizing the LSTM for temporal sequence analysis, the model can analyze the entire video sequence and make more informed predictions based on the context and progression of the video. These pre-processing steps lay the foundation for developing a robust and reliable deepfake detection model.

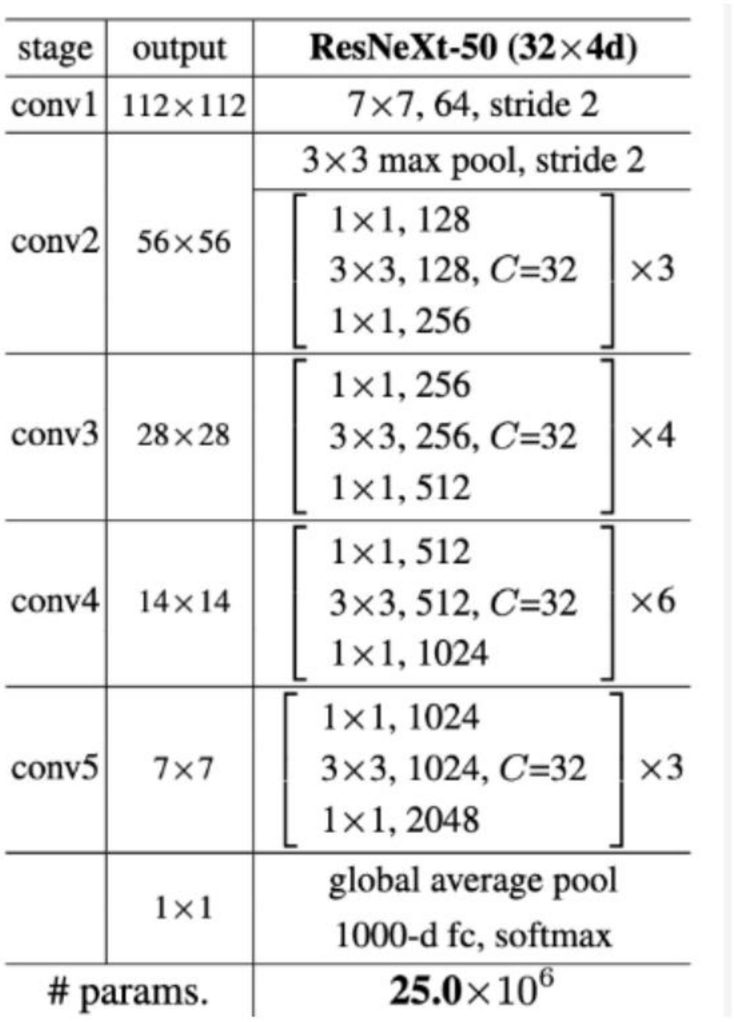
9.3.3 Model Details

The Model comprised of following layers

ResNext CNN

In the deepfake detection project, the Residual Convolution Neural Network (ResNext) pre-trained model is utilized to classify the videos as real or fake. ResNext is a variant of the popular Convolutional Neural Network (CNN) architecture, ResNext. The ResNext model consists of 50 layers and 32 x 4 dimensions, making it suitable for image classification tasks. The model has been pre-trained on a large dataset, which allows it to extract features from the input data effectively. The ResNext architecture utilizes the residual learning approach, which allows the model to learn more complex features by introducing skip connections between layers. These skip connections enable the model to learn the identity function more effectively, which is a significant factor in improving the model's performance.

The ResNext50_32x4d model used in our project has been fine-tuned on the dataset using transfer learning. Transfer learning is a machine learning technique that involves taking a pre-trained model and adapting it to a new task by fine-tuning the model's parameters on the new dataset. In the deepfake detection project, the pre-trained ResNext50_32x4d model is fine-tuned on the new dataset to improve its accuracy on the deepfake classification task. The fine-tuning process involves updating the model's weights using backpropagation and gradient descent, based on the loss computed during training. The fine-tuning process allows the model to adapt to the new dataset while retaining the features learned from the pre-training dataset, which results in improved accuracy on the deepfake classification task.



stage	output	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2
		$\begin{bmatrix}1\times1, 128\\ 3\times3, 128, C{=}32\\ 1\times1, 256\end{bmatrix}\times3$
conv3	28×28	$\begin{bmatrix}1\times1, 256\\ 3\times3, 256, C{=}32\\ 1\times1, 512\end{bmatrix}\times4$
conv4	14×14	$\begin{bmatrix}1\times1, 512\\ 3\times3, 512, C{=}32\\ 1\times1, 1024\end{bmatrix}\times6$
conv5	7×7	$\begin{bmatrix}1\times1, 1024\\ 3\times3, 1024, C{=}32\\ 1\times1, 2048\end{bmatrix}\times3$
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0**×10^6

Figure 18: ResNext Architecture

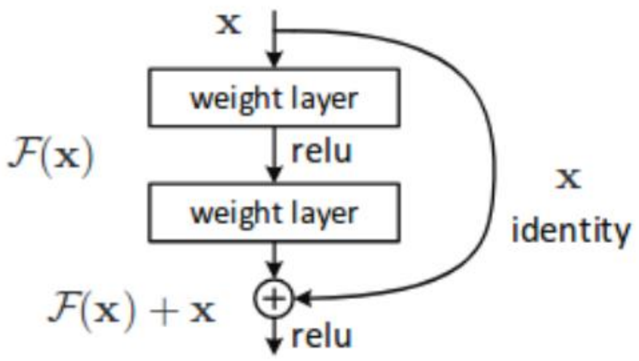


Figure 19: ResNext Working

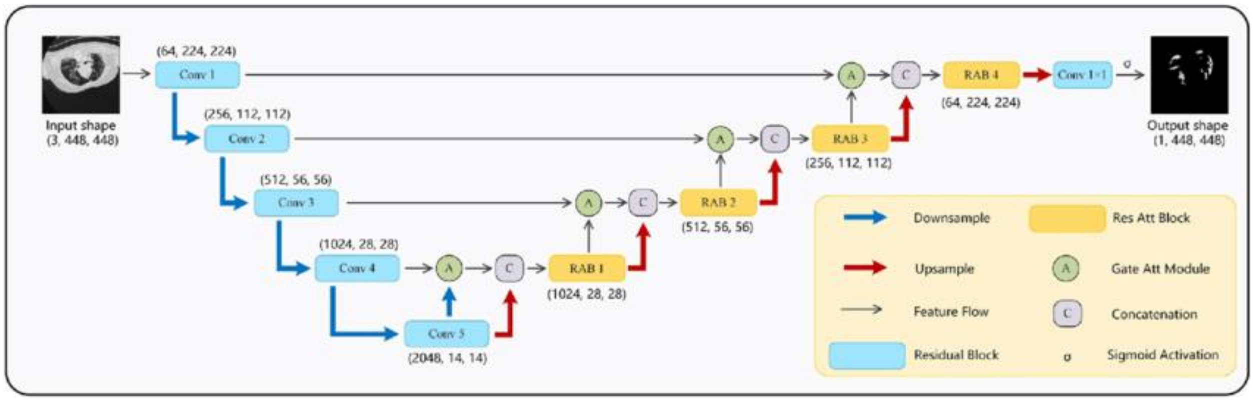


Figure 20: ResNeXt-50 (32 × 4d)

Sequential Layer

In the deep fake project, the Sequential layer plays a crucial role in storing the feature vector returned by the ResNext model in a sequential manner. This layer acts as a container for modules that can be stacked together and run simultaneously. The feature vector contains important information about the video frames that will be used for predicting whether the video is real or fake. The Sequential layer enables the feature vector to be passed to the LSTM sequentially, which performs temporal sequence analysis on the input data.

The Sequential layer provides a simple way of arranging the layers in the deep learning model. In the deep fake project, it allows the feature vector to be passed to the LSTM layer sequentially, which is essential for analyzing the temporal sequence of video frames. By using the Sequential layer, we can easily stack multiple layers of ResNext model and LSTM layer on top of each other, making it easy to create a complex deep learning model for detecting deep fake videos. The ordered sequence of feature vectors stored in the Sequential layer is passed through the LSTM layer, which helps to capture the temporal patterns in the video frames and makes the deep fake detection model more accurate.

LSTM Layer

The LSTM layer is a crucial component of the deepfake detection model as it is responsible for processing the sequence of feature vectors returned by the ResNext model. The model uses a single LSTM layer with 2048 latent dimensions and 2048 hidden layers. Additionally, a dropout of 0.4 is applied to prevent overfitting. The LSTM layer processes the frames of the video in a sequential manner to analyze the temporal changes between frames. By comparing the frames at 't' second with the frames of 't-n' seconds, where 'n' can be any number of frames before 't', the model is able to detect deepfakes based on changes in the video sequence.

The LSTM layer is particularly useful for detecting deepfakes as it is capable of analyzing the temporal changes between frames. This allows the model to detect deepfakes that have been created using advanced techniques that manipulate the video sequence to create realistic but fake videos. By comparing the frames at different time intervals, the model can identify patterns and inconsistencies that are indicative of deepfakes. Furthermore, the use of LSTM layer in the deepfake detection model allows for greater accuracy and reliability in detecting deepfakes, making it a valuable tool in the fight against the spread of misinformation and false information on social media platforms.

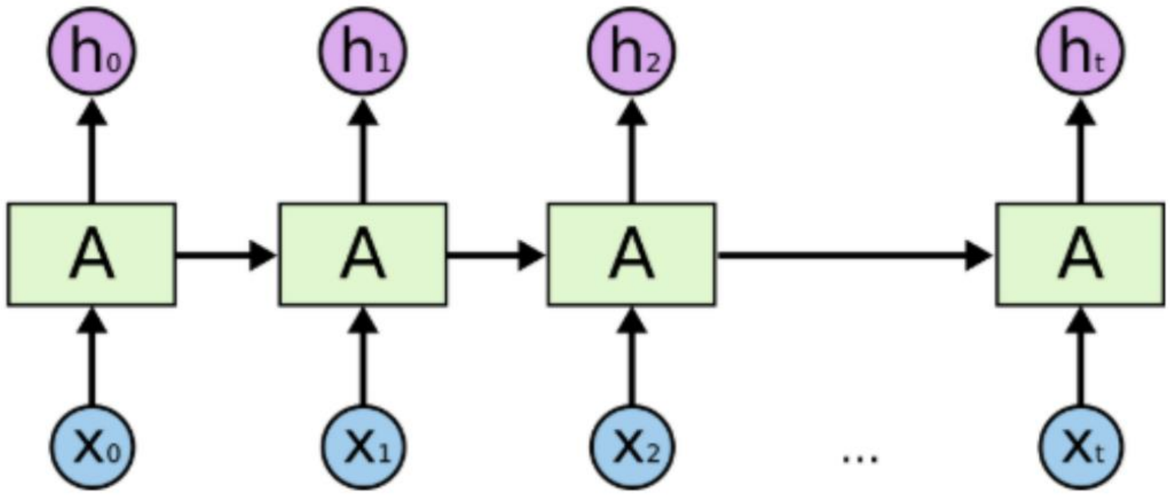


Figure 21: LSTM Architecture

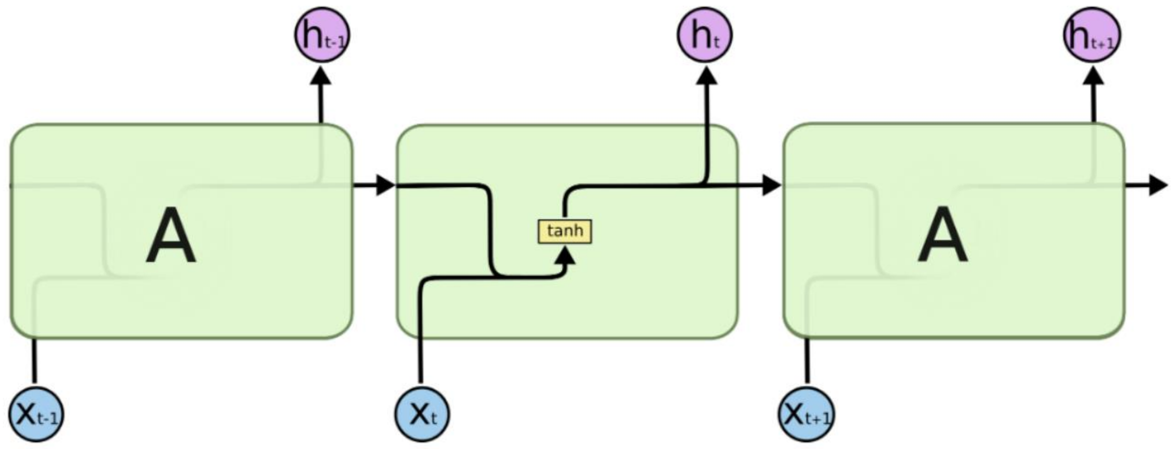


Figure 22: Internal LSTM Architecture

ReLU

In the deepfake detection project, ReLU activation function is used to introduce non-linearity into the neural network. ReLU stands for Rectified Linear Unit and it is a commonly used activation function in deep learning. ReLU activation function has several advantages over other activation functions. Firstly, it is computationally efficient and fast to compute. Secondly, it does not suffer from the vanishing gradient problem, which occurs with other activation functions like sigmoid and tanh. The vanishing gradient problem can cause issues with the training of deep neural networks. Additionally, ReLU has been shown to be more biologically plausible as it is similar to the way neurons in the brain work.

ReLU activation function is particularly useful in detecting deepfakes as it allows the neural network to better learn the non-linear features in the input data. In the case of deepfake detection, the neural network needs to be able to detect subtle changes in the video frames that may indicate that the video is fake. ReLU activation function helps the neural network to identify these non-linear features, which is crucial for the success of the deepfake detection system.

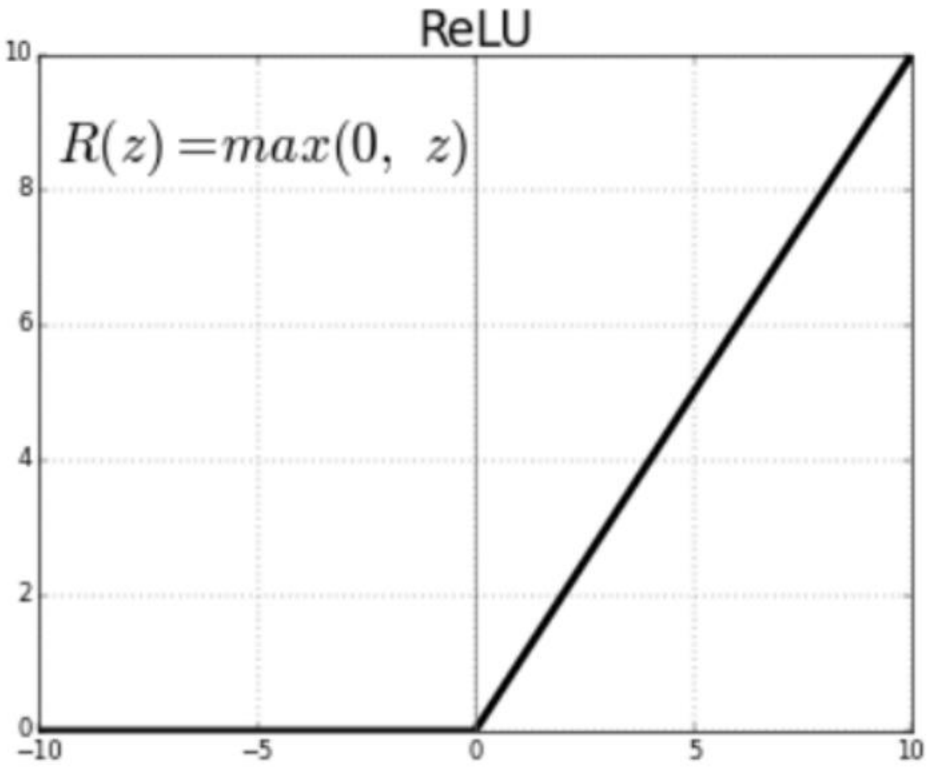


Figure 23: ReLU Activation

Dropout Layer

The use of dropout layer in the deepfake detection project is crucial to prevent overfitting, which occurs when a model becomes too complex and starts to memorize the training data instead of learning from it. By randomly setting the output for a given neuron to 0 during training, dropout layer encourages each neuron to be more independent and learn more robust features. This makes the model more capable of generalizing to new, unseen data, which is essential in detecting deepfakes that it has not seen before. Furthermore, by making the model more robust to noise and reducing the impact of individual neurons, dropout layer can also help prevent adversarial attacks that attempt to manipulate the model by introducing small perturbations to the input data.

In addition to its regularization effects, the use of dropout layer can also speed up the training process of the deepfake detection model. By reducing the interdependencies between neurons, dropout layer can make the optimization process less sensitive to the exact configuration of the network and hence more efficient. This means that the model can learn the same amount of information with fewer training epochs, which can save computational resources and time. Moreover, by enabling parallel processing of the input data, dropout layer can also take advantage of modern hardware architectures such as GPUs, which can significantly accelerate the training process.

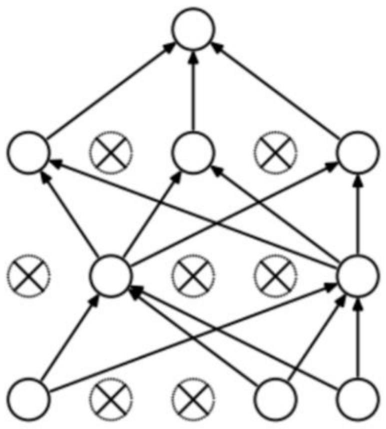


Figure 24: Dropout Layer Overview

Adaptive Average Pooling Layer

The Adaptive Average Pooling Layer is used in the deepfake detection model to perform pooling operation on the feature maps generated by the convolutional layers. Unlike the traditional max pooling or average pooling, adaptive average pooling performs pooling in a way that the output tensor size is fixed and not dependent on the input tensor size. This is achieved by using adaptive kernel sizes and stride values. This layer helps in reducing variance in the feature maps and extracting low-level features from the neighborhood, which can help in improving the model's performance.

The main advantage of using adaptive average pooling layer over traditional pooling layers is that it reduces computation complexity and makes the model more efficient. Additionally, adaptive average pooling layer is able to better capture spatial information in the feature maps, which is particularly important for tasks such as image classification and object detection.

9.3.4 Model Training Details

Train Test Split

The dataset used in this project is divided into two parts - training dataset and testing dataset. The training dataset consists of 70% of the total videos, which is equal to 4,200 videos, while the testing dataset consists of 30% of the videos, which is equal to 1,800 videos. The split between the two datasets is balanced, with 50% of the real videos and 50% of the fake videos in each dataset. A diagram is provided in Figure 7.6 for reference.

Data Loader

The Data Loader is a tool utilized for the purpose of loading videos and their corresponding labels, with a designated batch size of 4.

Training

The model was trained for a total of 20 epochs, utilising a learning rate of 1e-5 (0.00001), a weight decay of 1e-3 (0.001), and the Adam optimizer.

Adam Optimizer [32]

In order to adjust the learning rate during the training process, the model parameters are optimized using the Adam optimizer.

Cross Entropy

The loss function for this project is computed using the Cross Entropy approach, which is suitable for classification problems.

SoftMax Layer

In the deep fake project, the SoftMax layer is used as the final layer of the neural network model. This layer has two output nodes, representing the prediction of the video being either REAL or FAKE. The Softmax function is a squashing function that transforms the output of the neural network into probabilities. This is done by limiting the output of the function to a range between 0 and 1, which allows the output to be interpreted directly as a probability. Since the outputs of a

SoftMax function must sum up to 1, the Softmax layer is typically used as the final layer in neural networks.

The Softmax layer is crucial in the deep fake project as it provides us with the confidence or probability of the prediction. This allows us to interpret the results of the model and make informed decisions based on them. By using the Softmax layer as the final layer, we can effectively classify videos into two categories, REAL or FAKE, based on the probability distribution of the output nodes. This helps us in detecting deep fake videos with a high degree of accuracy and reliability, which is crucial in combating the spread of misinformation and ensuring the authenticity of digital media.

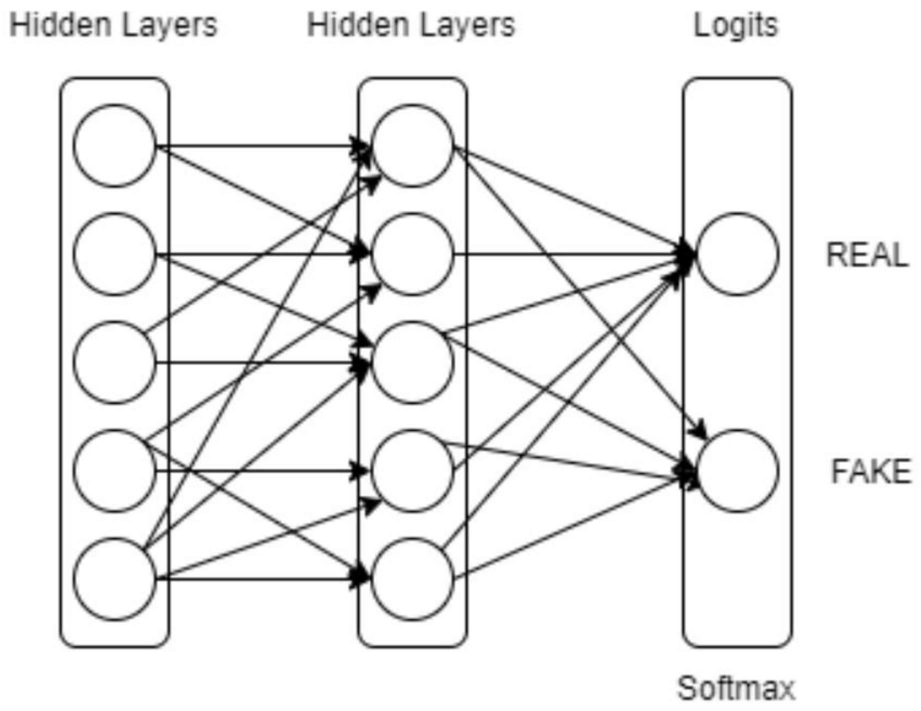


Figure 25: SoftMax Layer

Confusion Matrix

In the context of our project, a confusion matrix can help us evaluate the performance of our model in distinguishing between real and fake videos. The matrix would have four quadrants representing the four possible outcomes: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). A true positive is when the model correctly identifies a video as fake, while a true negative is when the model correctly identifies a video as real. False positive occurs when the model incorrectly identifies a real video as fake, while a false negative occurs when the model incorrectly identifies a fake video as real.

The confusion matrix can help us calculate various performance metrics such as accuracy, precision, recall, and F1 score. Accuracy is the percentage of correct predictions out of total predictions made by the model. Precision is the percentage of true positives out of all predicted positives, while recall is the percentage of true positives out of all actual positives. F1 score is the harmonic mean of precision and recall, providing a balanced measure of both. These metrics can help us understand the strengths and weaknesses of our model and fine-tune it accordingly for better performance.

Export Model

Once the training of the model is complete, the next step is to export the model so that it can be utilized for making predictions on real-time data. This involves saving the model architecture and its corresponding weights so that it can be loaded and used later without having to retrain the entire model. By exporting the model, we can make use of it in various applications where we need to classify videos as real or fake in real-time.

9.3.5 Model Prediction Details

In the application, we load the pre-trained model. Then, when a new video needs to be predicted, it goes through preprocessing steps (refer to Section 8.2.2 and Section 9.3.2) and is passed to the loaded model. The model performs the prediction on the video, and the output is whether the video is real or fake, along with the level of confidence in the prediction.

Chapter 10: Deployment And Maintenance

10.1 Deployment

We follow these steps:

1. We have created a Docker image of the web application using React and PyTorch libraries.

2. The Docker image was then pushed to the Docker Hub repository.

3. We have created an Amazon Machine Image (AMI) of Ubuntu 20.04 LTS on an EC2 instance.

4. An EC2 instance was launched using the previously created AMI.

5. Once the instance was running, we have installed the necessary libraries and dependencies, including Docker and AWS CLI.

6. The Docker image was then pulled from the Docker Hub repository to the EC2 instance.

7. We have created a script that runs the Docker container and starts the React application.

8. An Elastic Load Balancer (ELB) was created to distribute incoming traffic across the EC2 instances.

9. The EC2 instances were launched in a private subnet and the ELB was placed in a public subnet.

10. Finally, the security groups were configured to allow traffic to and from the ELB and the EC2 instances.

This deployment process allows for easy scaling and management of the web application while maintaining security and reliability.

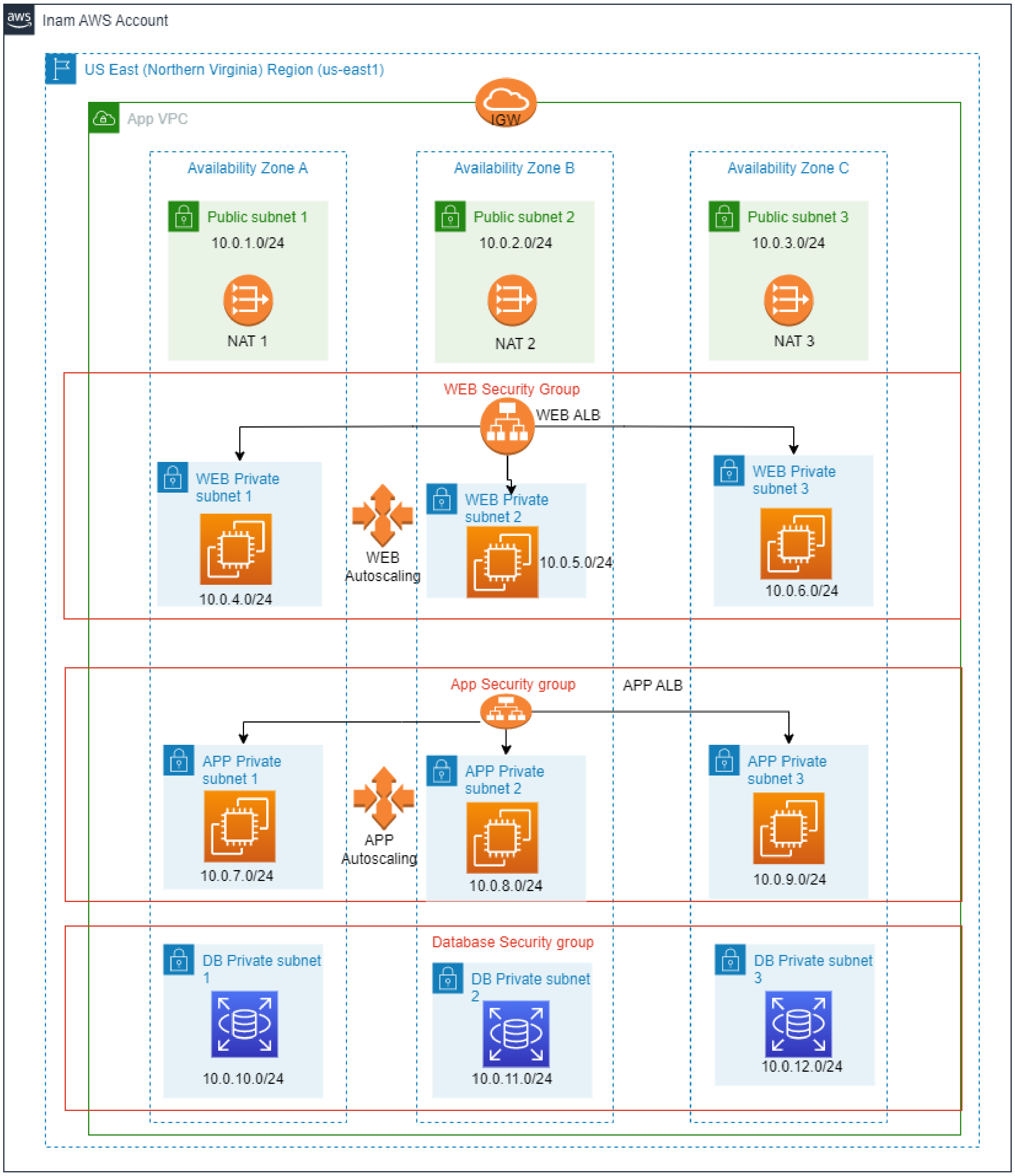


Figure 26: AWS Application Architecture

The architecture includes Amazon Virtual Private Cloud (Amazon VPC), Amazon Elastic Compute Cloud (Amazon EC2), Amazon Relational Database Service (Amazon RDS) with high availability, and Elastic Load Balancing (ELB). The network traffic flows seamlessly through the different AWS components - from the client to the backend database, and back to the client. With the use of NAT in the public subnet and ELB in the private subnet, the architecture provides an additional layer of security and isolation between the internet and the backend resources. The NAT gateway acts as a gateway for all traffic going in and out of the private subnet, and the ELB ensures that traffic is distributed evenly across the Amazon EC2 instances. The Amazon RDS database provides a scalable and managed database service that can handle the data processing requirements of the application. Overall, this architecture provides a highly available, scalable, and fault-tolerant solution for hosting our web application on AWS.

Steps

Following are the steps to be followed for the deployment of the application.

1. clone the repository using the below command.

git clone https://github.com/Muhammad-Musab/medifor

Note: As its a private repository only authorized users will be able see the code

and do the process of deployment

2. docker build -t mediaforensic:v1 .

3. docker run -dp 0.0.0.0:8000 mediaforensic:v1

10.2 Maintenance

Maintenance is an essential aspect of any system to ensure its smooth operation and to prevent any issues. The following are some of the maintenance processes that can be applied to our Deepfake detection project:

1. Regular updates: Keeping the system updated with the latest software and security patches is crucial to prevent any vulnerabilities. This includes updating the operating system, frameworks, and libraries used in the project.

2. Monitoring: Continuous monitoring of the system's performance and logs is important to identify any potential issues and to ensure the system is running smoothly. Any unusual activity should be investigated and resolved promptly.

3. Backups: Regular backups of the data and system configuration are necessary to prevent data loss and to ensure business continuity. Backups can be automated and stored in a secure location.

4. Testing: Regular testing of the system's functionality and performance is important to ensure that it is operating as expected. This includes testing the system's response to different scenarios, load testing, and stress testing.

5. Security: Regular security audits and assessments should be conducted to identify and address any security risks. This includes reviewing access controls, network security, and authentication mechanisms.

By implementing these maintenance processes, we can ensure the system's availability, reliability, and security. Regular maintenance can prevent downtime, improve system performance, and enhance the user experience.

Steps

Following are the steps to be followed for updating the code to the latest version

of the application.

1. docker stop mediaforensic:v1

2. git pull

Note: As it's a private repository only authorized users will be able to see the code

and do the process of deployment

3. Copy all the files and trained models into the project folder(optional: Do this if any new files and

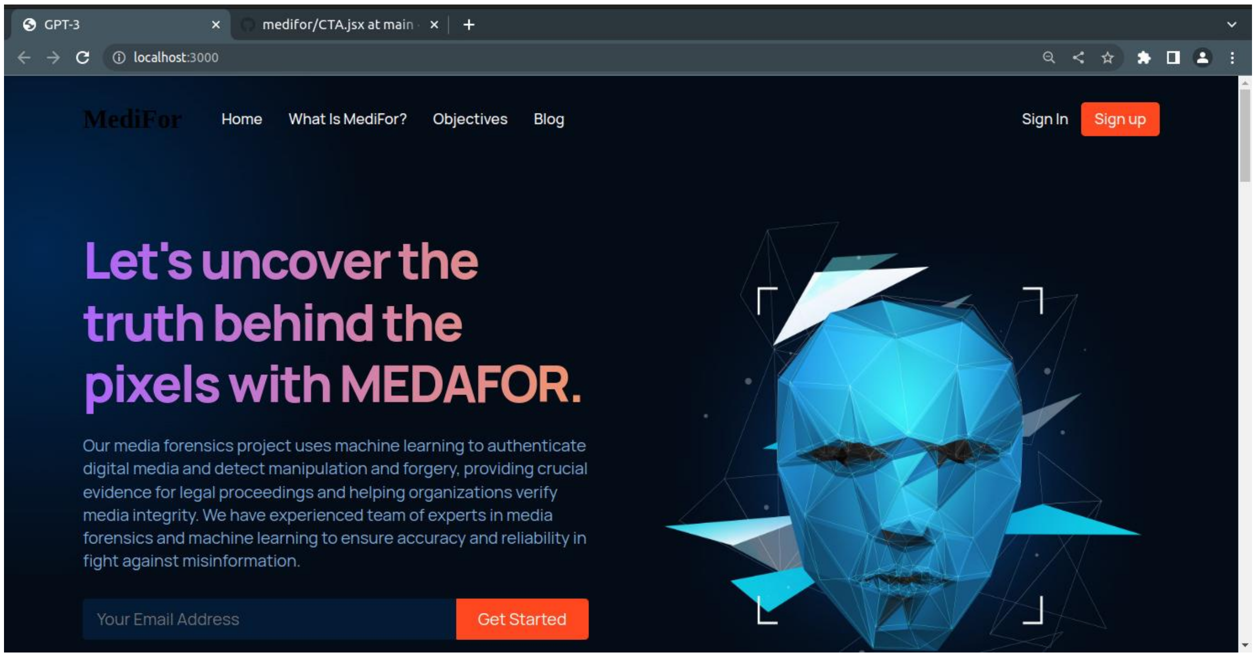
models are added).

4. docker build -t mediaforensic:latest .

5. docker run -dp 0.0.0.0:8000 mediaforensic:latest

Chapter 11: Results and Discussions

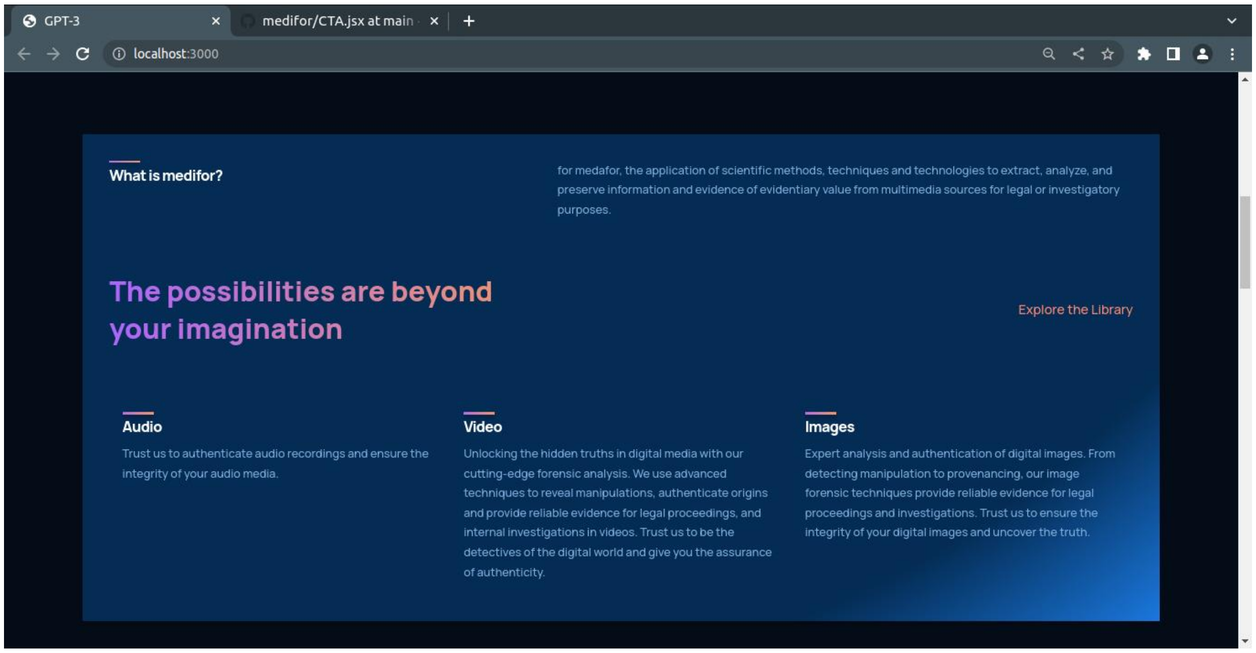
11.1 Screenshots



Let's uncover the truth behind the pixels with MEDAFOR.

Our media forensics project uses machine learning to authenticate digital media and detect manipulation and forgery, providing crucial evidence for legal proceedings and helping organizations verify media integrity. We have experienced team of experts in media forensics and machine learning to ensure accuracy and reliability in fight against misinformation.

Figure 27: Web App



What is medifor?

for medafor, the application of scientific methods, techniques and technologies to extract, analyze, and preserve information and evidence of evidentiary value from multimedia sources for legal or investigatory purposes.

The possibilities are beyond your imagination

Audio

Trust us to authenticate audio recordings and ensure the integrity of your audio media.

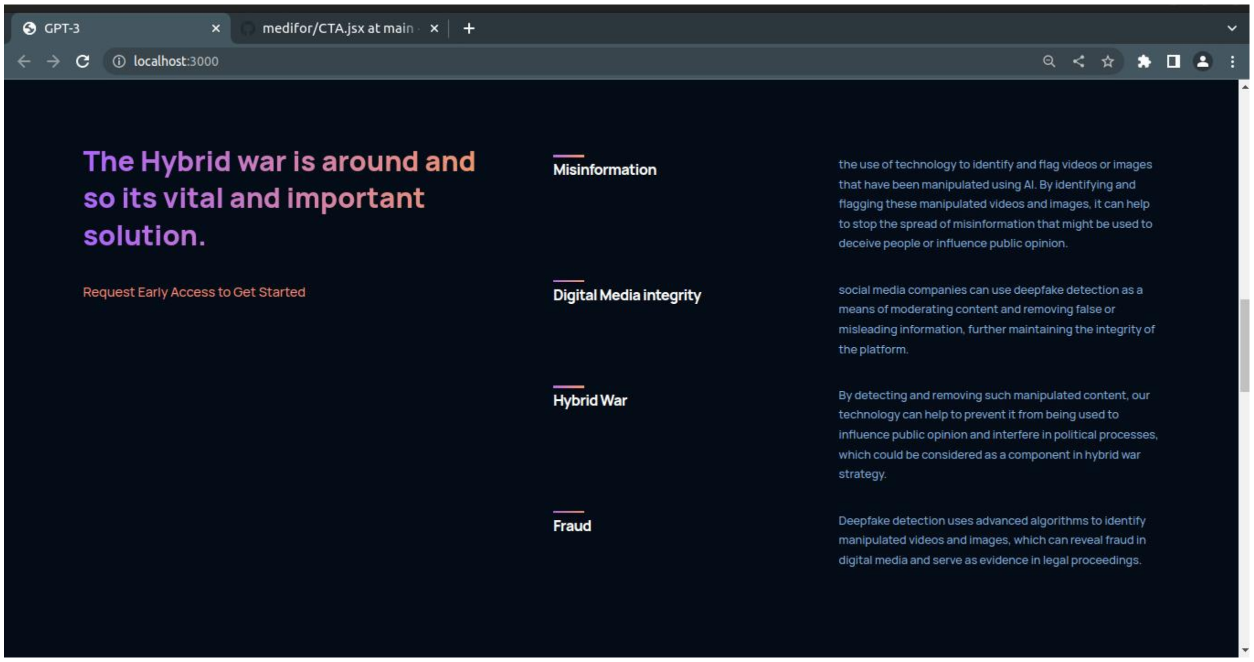
Video

Unlocking the hidden truths in digital media with our cutting-edge forensic analysis. We use advanced techniques to reveal manipulations, authenticate origins and provide reliable evidence for legal proceedings, and internal investigations in videos. Trust us to be the detectives of the digital world and give you the assurance of authenticity.

Images

Expert analysis and authentication of digital images. From detecting manipulation to provenancing, our image forensic techniques provide reliable evidence for legal proceedings and investigations. Trust us to ensure the integrity of your digital images and uncover the truth.

Figure 28: Web App



The Hybrid war is around and so its vital and important solution.

Request Early Access to Get Started

Misinformation

the use of technology to identify and flag videos or images that have been manipulated using AI. By identifying and flagging these manipulated videos and images, it can help to stop the spread of misinformation that might be used to deceive people or influence public opinion.

Digital Media integrity

social media companies can use deepfake detection as a means of moderating content and removing false or misleading information, further maintaining the integrity of the platform.

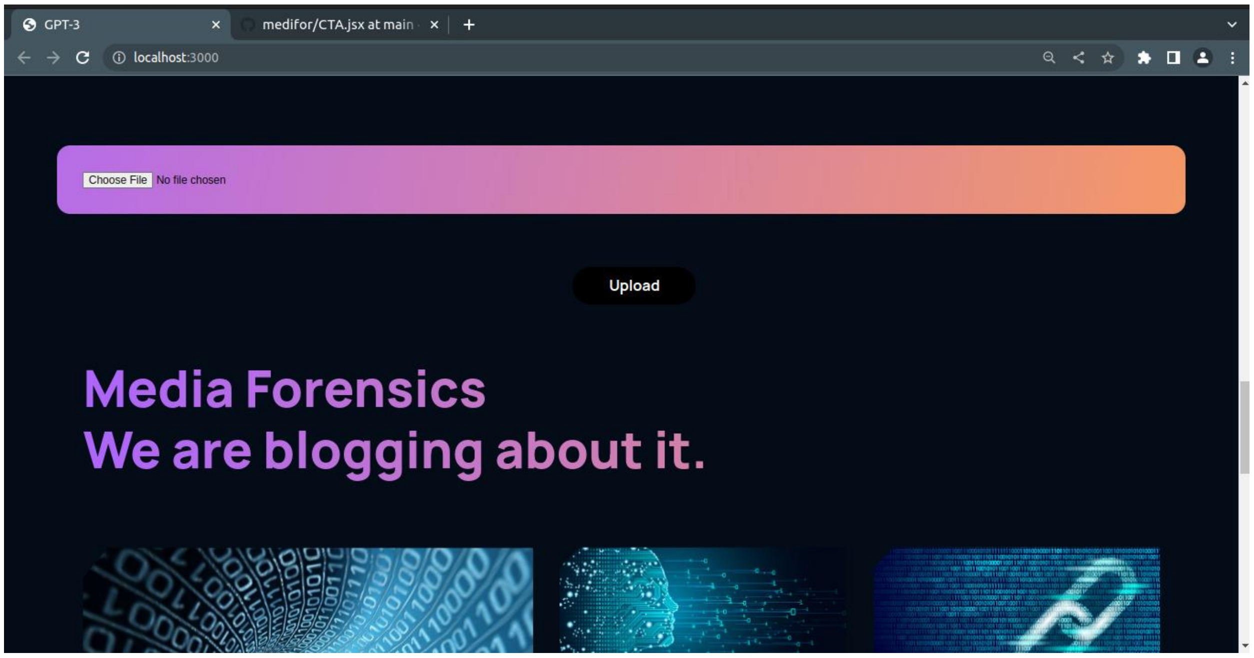
Hybrid War

By detecting and removing such manipulated content, our technology can help to prevent it from being used to influence public opinion and interfere in political processes, which could be considered as a component in hybrid war strategy.

Fraud

Deepfake detection uses advanced algorithms to identify manipulated videos and images, which can reveal fraud in digital media and serve as evidence in legal proceedings.

Figure 29: Web App



Media Forensics
We are blogging about it.

Figure 30: Web App

11.2 Model Results

Dataset	No. of videos	Sequence Length	Accuracy
FaceForensic	2000	20	90.95
DFDC+CelebDF+FF++	6000	20	87.79

Table 3: Model Results

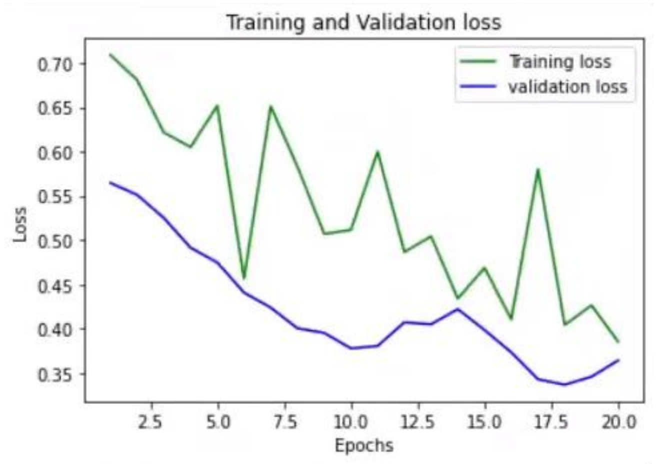


Figure 31: Training and Validation Loss

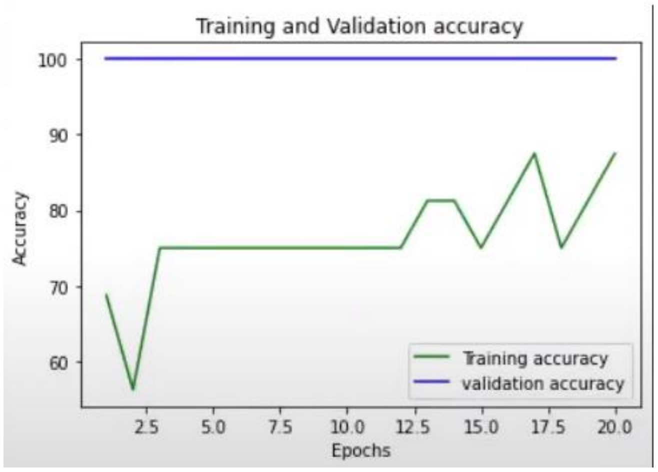


Figure 32: Training and Validation Accuracy

11.3 Discussion on Results

The results of our deepfake project have been quite promising. We trained our model on the Face Forensic++ dataset with 20 sequences and achieved an accuracy of 90.95%. Additionally, we tested our model on a mixed dataset of real and fake videos from DFDC, FF++ and Celeb DF, and achieved an accuracy of 87% with the same number of sequences. These results indicate that our model is capable of distinguishing between real and fake videos with a high level of accuracy.

We used a combination of CNN and LSTM layers to train our model. The CNN layers were used for feature extraction from individual frames, while the LSTM layer processed the frames sequentially to analyze the temporal changes in the video. Additionally, we used ReLU activation function in our model, which is closer to the way our biological neurons work and has the advantage of not having any backpropagation errors unlike the sigmoid function. We also used a Dropout layer with a value of 0.4 to avoid overfitting in the model.

The increase in training accuracy and decrease in training loss indicate that the model is learning the patterns in the data and improving its ability to classify between real and fake videos. However, it is important to note that overfitting can occur if the model becomes too complex and starts to memorize the training data instead of learning the underlying patterns.

To address this issue, a dropout layer with a value of 0.4 was added to the model to avoid overfitting. Dropout randomly sets the output for a given neuron to 0 during training, which forces the model to learn more robust features that generalize better to new data. The addition of the dropout layer also helps to make the cost function more sensitive to neighboring neurons, changing the way the weights are updated during backpropagation. Overall, the use of dropout helps to improve the model's ability to generalize and perform well on new, unseen data.

To evaluate the performance of our model, we used a confusion matrix which showed the ways in which our classification model is confused when it makes predictions. The confusion matrix gave us insight into the types of errors that were being made. For example, the false positive rate was higher than the false negative rate. This means that our model was more likely to classify a real video as fake than the other way around.

Chapter 12: Conclusion and Future Scope

12.1 Conclusion

In conclusion, we are successful in developing a neural network model that can accurately distinguish between real and fake videos. The use of convolutional neural networks along with techniques like ReLU activation, dropout layers, and adaptive average pooling layers helped improve the accuracy and reduce overfitting. The model was trained on the FaceForensic++, DFDC and CelebDF, a mixed dataset and achieved high accuracy rates of 90% and 87% respectively. These results demonstrate the potential of using machine learning for detecting deepfakes.

This project highlights the potential of machine learning and deep learning techniques in detecting deepfakes and offers a promising approach to mitigating the risks associated with this technology. As deepfake technology continues to evolve and become more sophisticated, the development of effective detection methods will be crucial in preserving the integrity of digital media and combating the spread of misinformation.

12.2 Future Scope

Any existing system may be made better, and this is especially true with future-facing projects built with cutting-edge technologies.

- The web-based platform can be upscaled to a browser plugin for ease of access to the user.

- Currently, only face deep fakes are being detected by the algorithm, but the algorithm can be enhanced to detect full-body deep fakes.

- The application can be deployed in a free cloud for better accessibility.

- Creating an open-source API for detection can make the technology more accessible to other developers.

- Batch processing of entire video instead of processing first 'x' frames can be implemented to enhance the accuracy.

- Optimizing the code for faster execution can help to reduce processing time and increase efficiency.

- More diverse datasets can be used to train the model, resulting in better accuracy and reliability.

- Research can be done to incorporate real-time detection and prevention of deep fakes in social media platforms.

Chapter 13: Appendix

References/ Bibliography

[1] Face app: https://www.faceapp.com/ (Accessed on 26 April, 2023)

[2] Face Swap : https://faceswaponline.com/ (Accessed on 26 April, 2023)

[3] perov, I. (2020) Iperov/deepfacelab: Deepfacelab is the leading software for creating deepfakes., GitHub. Available at: https://github.com/iperov/DeepFaceLab (Accessed: April 24, 2023).

[4] Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Mr. Dpfks, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, Pingyu Wu, Bo Zhou, Weiming Zhang "DeepFaceLab: Integrated, flexible and extensible face-swapping framework" in arXiv:2005.05535

[5] Kelley M. Tayler; Laurie A. Harris (8 June 2021). Deep Fakes and National Security (Report). Congressional Research Service. p. 1. Retrieved 19 July 2021.

[6] Roettgers, Janko (21 February 2018). "Porn Producers Offer to Help Hollywood Take Down Deepfake Videos". *Variety*. Archived from the original on 10 June 2019. Retrieved 28 February 2018.

[7] Dickson, E. J. (7 October 2019). "Deepfake Porn Is Still a Threat, Particularly for K-Pop Stars". *Rolling Stone*. Archived from the original on 30 October 2019. Retrieved 9 November 2019.

[8] "The State of Deepfake - Landscape, Threats, and Impact" (PDF). *Deeptrace*. 1 October 2019. Archived (PDF) from the original on 9 August 2020. Retrieved 7 July 2020.

[9] Roettgers, Janko (21 February 2018). "Porn Producers Offer to Help Hollywood Take Down Deepfake Videos". *Variety*. Archived from the original on 10 June 2019. Retrieved 28 February 2018.

[10] Romano, Aja (18 April 2018). "Jordan Peele's simulated Obama PSA is a double-edged warning against fake news". *Vox*. Archived from the original on 11 June 2019. Retrieved 10 September 2018.

[11] Parker, Ashley (7 September 2020). "Trump and allies ramp up efforts to spread disinformation and fake news". *The Independent*. Retrieved 9 April 2022.

[12] Allyn, Bobby (16 March 2022). "Deepfake video of Zelenskyy could be 'tip of the iceberg' in info war, experts warn". NPR. Retrieved 17 March 2022.

[13] Sarah Cahlan "How misinformation helped spark an attempted coup in Gabon" https://www.washingtonpost.com/politics/2020/02/13/how-sick-president-suspect-video-helped-sparked-an-attempted-coup-gabon/

[14] "Deepfake Putin is here to warn Americans about their self-inflicted doom". *MIT Technology Review*. Archived from the original on 30 October 2020. Retrieved 7 October 2020

[15] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies,Matthias Nießner, "FaceForensics++: Learning to Detect Manipulated Facial Images" in arXiv:1901.08971.

[16] Deepfake detection challenge dataset : https://www.kaggle.com/c/deepfake-detection challenge/data Accessed on 26 March, 2020

[17] Yuezun Li , Xin Yang , Pu Sun , Honggang Qi and Siwei Lyu "Celeb-DF: A Large-scale Challenging Dataset for DeepFake Forensics" in arXiv:1909.12962

[18] Yuezun Li, Siwei Lyu, "ExposingDF Videos By Detecting Face Warping Artifacts," in arXiv:1811.00656v3.

[19] Huy H. Nguyen , Junichi Yamagishi, and Isao Echizen " Using capsule networks to detect forged images and videos " in arXiv:1810.1121

[20] Mitra, A., Mohanty, S.P., Corcoran, P. et al. A Machine Learning Based Approach for Deepfake Detection in Social Media Through Key Video Frame Extraction. SN COMPUT. SCI. 2, 98 (2021). https://doi.org/10.1007/s42979-021-00495-x

[21] D. Güera and E. J. Delp, "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Auckland, New Zealand, 2018, pp. 1-6.

[22] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8, June 2008. Anchorage, AK

[23] N. Bonettini, E. D. Cannas, S. Mandelli, L. Bondi, P. Bestagini and S. Tubaro, "Video Face Manipulation Detection Through Ensemble of CNNs," 2020 25th International Conference on Pattern Recognition (ICPR), 2021, pp. 5012-5019, doi: 10.1109/ICPR48806.2021.9412711.

[24] Sensity https://sensity.ai/

[25] Quantum Integrity https://quantumintegrity.ch/

[26] Deepware Scanner https://github.com/deepware

[27] The Sentinel https://thesentinel.ai/

[28] Deepfake-o-Meter https://zinc.cse.buffalo.edu/ubmdfl/deep-o-meter/

[29] G. L. Team, "My Great Learning," Great Lakes E-Learning Services Pvt. Ltd, 2022. [Online]. Available: https://www.mygreatlearning.com/blog/all-you-need-to-know-about-deepfake-ai/#:~:text=Deepfake%20content%20is%20created%20by,content%20is%20real%20or%20artificial..

[30] Generative Adversarial Networks: The Future of Deepfakes

https://blog.metaphysic.ai/the-future-of-generative-adversarial-networks-in-deepfakes/

[31] Face swap Picture: https://clippingpanda.com/wp-content/uploads/2020/10/Photoshop-Face-Swap.jpg

[32] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. arXiv:1412.6980, Dec. 2014.

[33] 10 deepfake examples that terrified and amused the internet : https://www.creativebloq.com/features/deepfake-examples Accessed on 26 March, 2020

[34] J. Thies et al. Face2Face: Real-time face capture and reenactment of rgb videos. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2387–2395, June 2016. Las Vegas, NV

[35] Deepfakes, Revenge Porn, And The Impact On Women : https://www.forbes.com/sites/chenxiwang/2019/11/01/deepfakes-revenge-porn-andthe-impact-on-women

[36] Deepfakes and fake news pose a growing threat to democracy, experts warn: https://news.northeastern.edu/2022/04/01/deepfakes-fake-news-threat-democracy/

Project Timeline

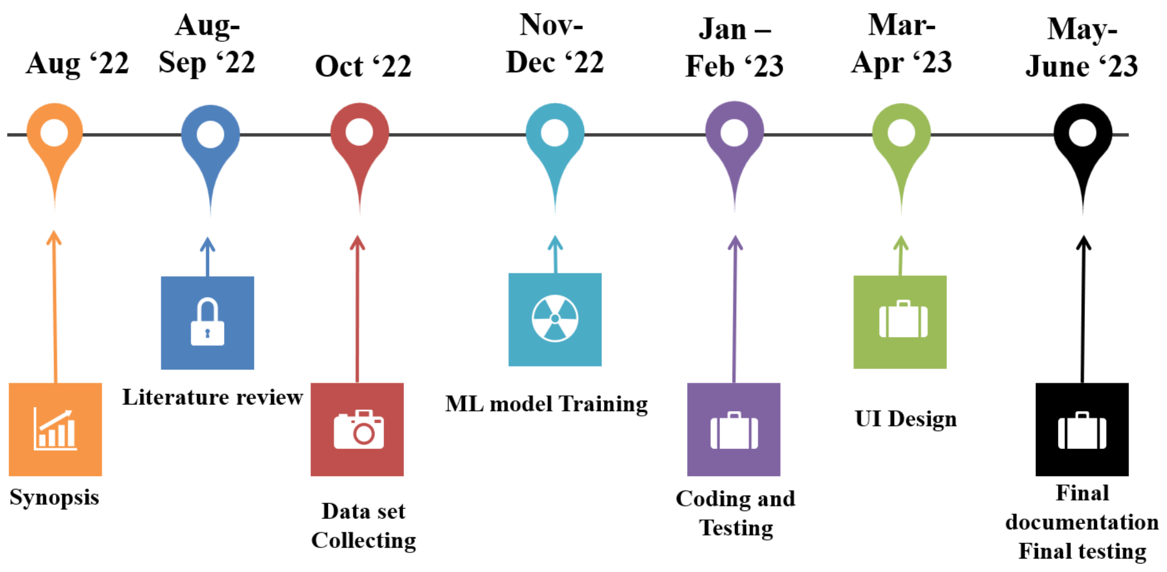
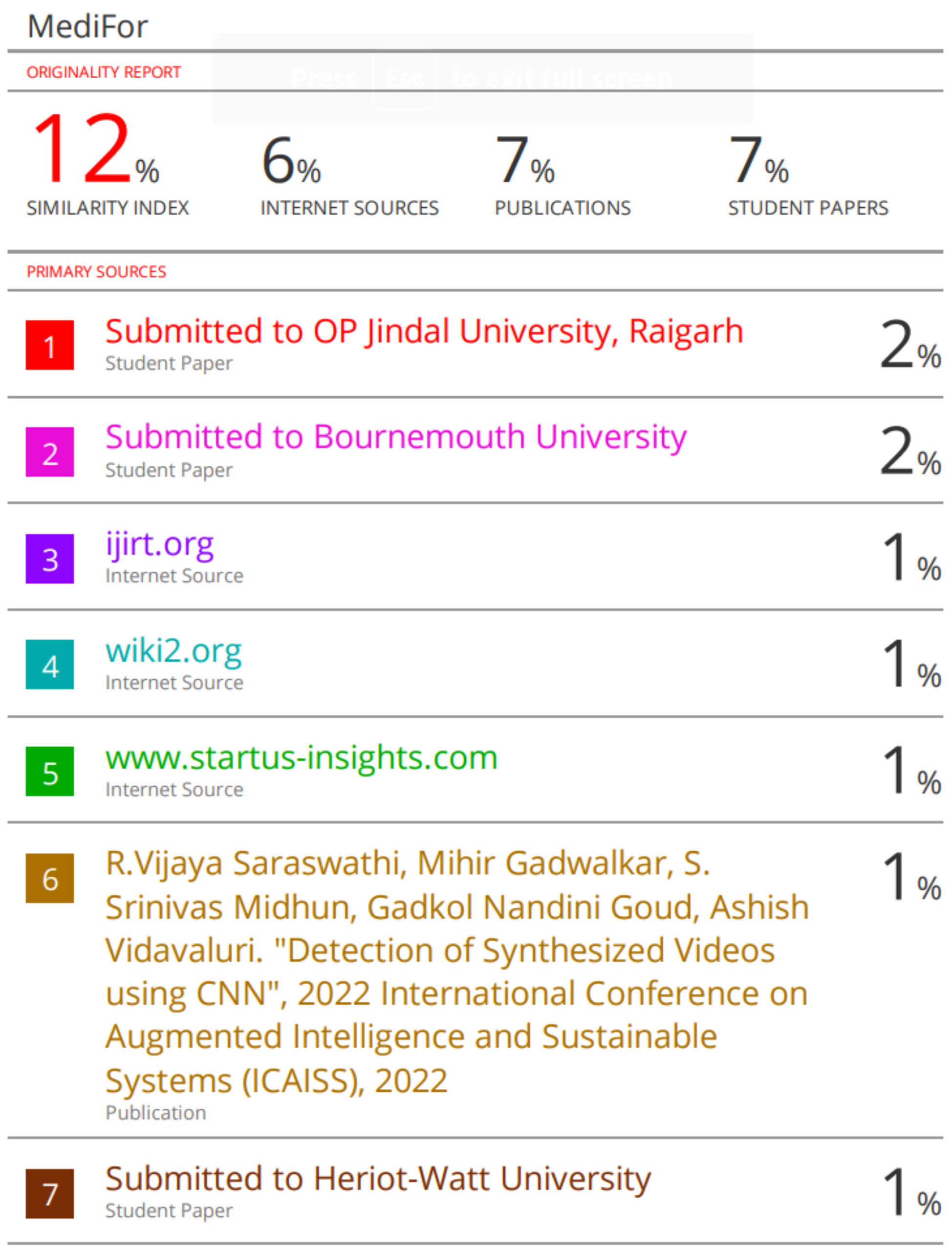


Figure 33: Project Timeline

Plagiarism Report



MediFor

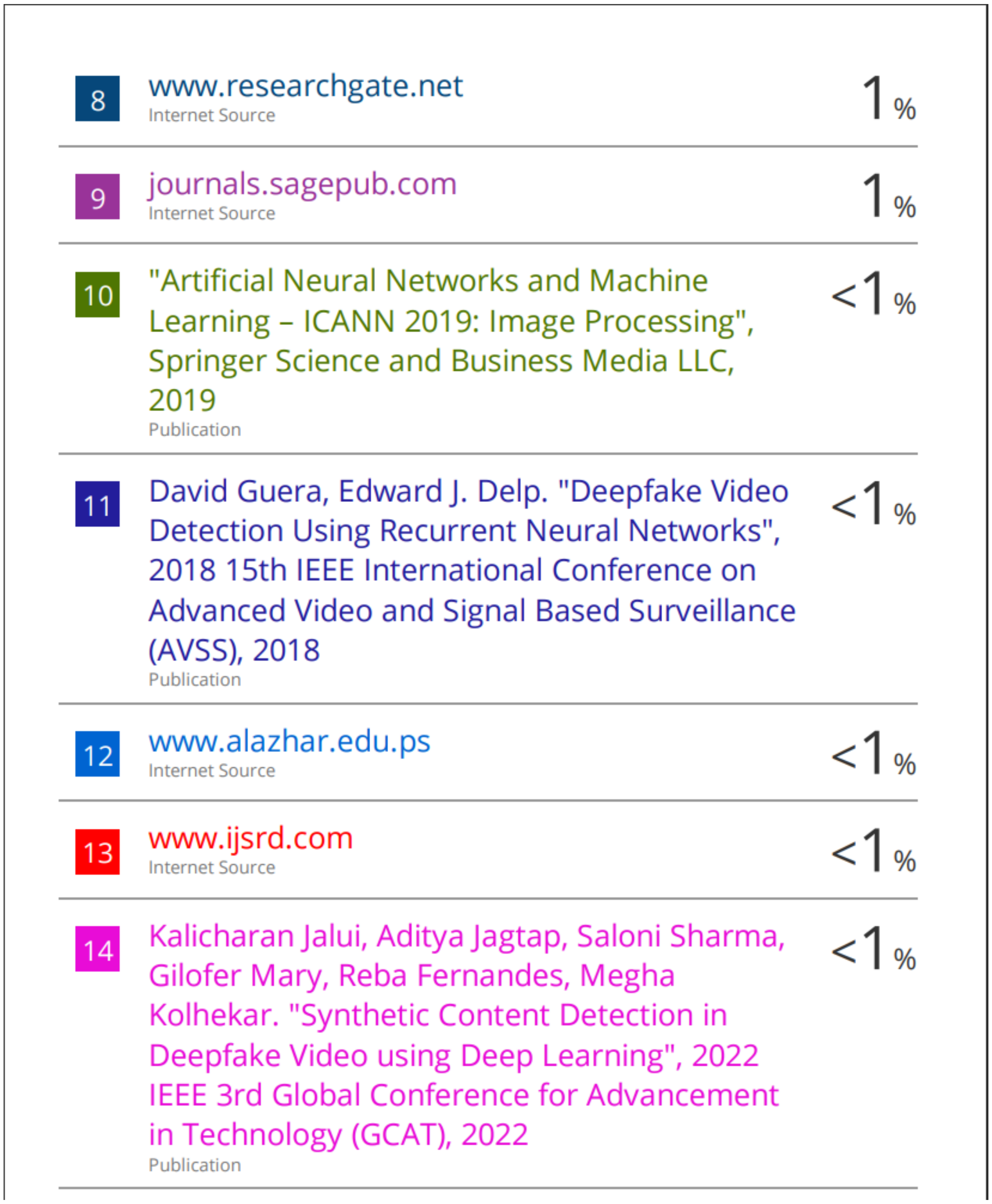
Press Esc to exit full screen

ORIGINALITY REPORT

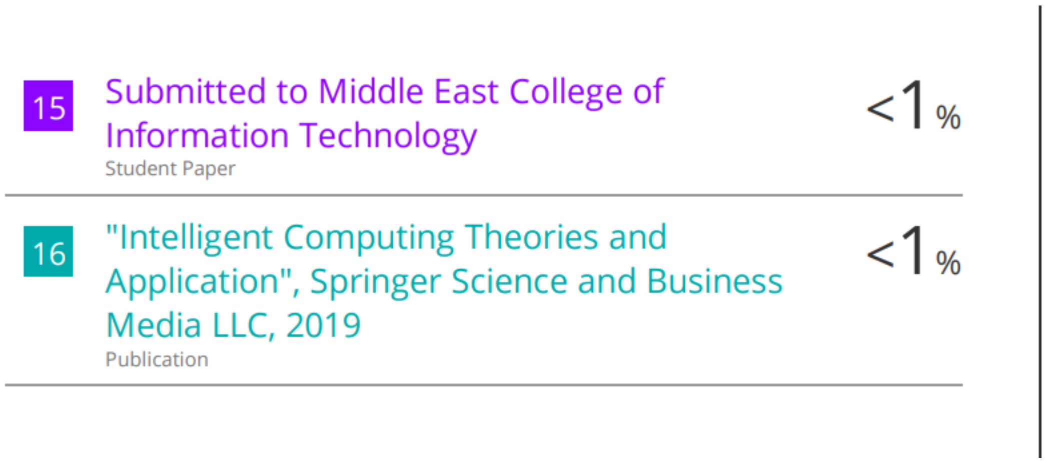
12%	6%	7%	7%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to OP Jindal University, Raigarh Student Paper	2%
2	Submitted to Bournemouth University Student Paper	2%
3	ijirt.org Internet Source	1%
4	wiki2.org Internet Source	1%
5	www.startus-insights.com Internet Source	1%
6	R.Vijaya Saraswathi, Mihir Gadwalkar, S. Srinivas Midhun, Gadkol Nandini Goud, Ashish Vidavaluri. "Detection of Synthesized Videos using CNN", 2022 International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), 2022 Publication	1%
7	Submitted to Heriot-Watt University Student Paper	1%



8	www.researchgate.net Internet Source	1 %
9	journals.sagepub.com Internet Source	1 %
10	"Artificial Neural Networks and Machine Learning – ICANN 2019: Image Processing", Springer Science and Business Media LLC, 2019 Publication	<1 %
11	David Guera, Edward J. Delp. "Deepfake Video Detection Using Recurrent Neural Networks", 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018 Publication	<1 %
12	www.alazhar.edu.ps Internet Source	<1 %
13	www.ijsrd.com Internet Source	<1 %
14	Kalicharan Jalui, Aditya Jagtap, Saloni Sharma, Gilofer Mary, Reba Fernandes, Megha Kolhekar. "Synthetic Content Detection in Deepfake Video using Deep Learning", 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT), 2022 Publication	<1 %



15	Submitted to Middle East College of Information Technology Student Paper	<1%
16	"Intelligent Computing Theories and Application", Springer Science and Business Media LLC, 2019 Publication	<1%

Meeting Logs

FYP Meetings Log

S.No	**Discussion Points**	**Date**	**Supervisor Signature**
1	Discussed About CASIA and CASIA v2.0 image datasets.CASIA v2.0 is an updated version with more advanced editing and improved annotation. Both datasets are divided into training, validation, and test sets.	16/11/2022	
2	Discussed About NIST(Data Set) is the National Institute of Standards and Technology, a non-regulatory agency of the U.S. Department of Commerce. NIST maintains a number of datasets that are commonly used in research, including datasets of handwritten characters, fingerprints, and facial images. These datasets are used in a variety of research areas, including pattern recognition, machine learning, and biometrics.	23/11/2022	
3	Discussed about Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs) are two popular machine learning algorithms that are used in many different applications. CNNs are a type of neural network that are particularly well-suited to image classification tasks. SVMs are a type of algorithm that can be used for classification, regression, and outlier detection.	14/12/2022	
4	Discussed about The DFDC dataset includes a large number of deepfake and real videos, which can be used to train and test deepfake detection algorithms.Like the DFDC dataset, the FF++ dataset is intended to support research on detecting and combating the use of deepfake and other manipulated media.	21/12/2022	
5	Discussed about ReXNet (short for "Residual Extremely Sparse Convolutional Neural Network") and LTSM (short for "Long Short-Term Memory") is a type of recurrent neural network (RNN) for video classification	4/01/2023	
6	Model Training Details. The Data preparation details and model architecture design	09/01/2023	
7	Discussed about Hyperparameter tuning, Evaluating the model, fine-tuning and testing	27/01/2023	

8	Discussed about hardware issues: This can occur when the system running the training process is not powerful enough or does not have enough memory to handle the size of the model or dataset.	08/02/2023	
9	Showed the result and accuracy that we achieved after training the models	15/02/2023	
10	Discussed about AWS and DevOps for final Deployment And Maintenance.	08/03/2023	